Administrator’s Guide

Version 7.1.4
Note
Before using this information and the product it supports, read the information in Appendix C, “Notices”, on page 319.
Contents

About This Book ........................................ vii
Who Should Read This Guide .......................... vii
What This Guide Contains ............................. vii
Publications ............................................. viii
IBM Tivoli NetView for UNIX Library .......... viii
Related publications ................................. viii
Accessing publications online ..................... viii
Ordering publications ............................... ix
Contacting software support ...................... ix
Typeface Conventions ............................... x
Accessibility Information ............................ x
Keyboard Access ....................................... x
Participating in newsgroups ....................... x

Chapter 1. Understanding Tivoli NetView
Processes ............................................. 1

Foreground Processes ................................. 1
  ovw Application ................................ 2
  ipmap Application ............................... 2
  xmap Application ................................ 3
  nvevents Application ............................. 3
  xnmtrap Application .............................. 4
  xnmnsmpconf Application ......................... 4
  xnmloadmib Application .......................... 4
  Web Console MIB Loader Application .......... 4
  Web Console MIB Browser Application ........ 4
  xnmbrowser Application ........................ 4
  xnmcollect Application ........................ 5
  xnmbuilder Application ........................ 5
  xnmgraph Application ........................... 5
  xnmfault Application ............................ 5
  xnmrunreport Application ......................... 5
  nmpolling Application ........................... 5
  backup Application ................................ 6
  shpmun Application ............................... 6
  xnmappmon Application ......................... 6
  Web-Based Device Management .................. 6

Process Management ................................... 7
  The ovspmd Daemon ................................ 7

Background Processes ................................. 7
  Topology Discovery and Database Daemons ...... 7
  Event and Trap Processing Daemons ............. 11
  Security Services Daemons ....................... 15
  Service Discovery and Monitoring Daemon ...... 15
  Host Connection Daemons (AIX Only) ............ 16
  Using First Failure Data Capture to Resolve Daemon Problems .................. 16
  Working With the FFDC Function ............... 16

Databases ............................................. 17
  Map Database .................................. 17
  Object Database ................................. 17
  IP Topology Database ............................. 18
  General Topology Manager Database .......... 18
  Object Registration Service Database .......... 18

Customizing and Generating Reports on the Object Database ........................ 18
  The nvdbimport Utility ........................... 18
  The nvdbformat Utility ........................... 19
  Role of the SEAs ................................. 19

Chapter 2. Defining and Managing a
Security Policy ........................................ 21

Understanding Tivoli NetView Security Services .... 21
  Network Authentication and Identification . ... 21
  Protected Network Communication .............. 22
  Log In Process .................................... 22
  Log In Considerations ............................ 22
  Password Protection ............................. 23
  Continuous, Auditable Network Management .... 23
  Network Access Control of Tivoli NetView Resources .......................... 24
  Customized Tivoli NetView Graphical Interface 25
  Audit Management ................................ 25
  Consistent Security Controls .................... 25
  Pager Service For Event Correlation Rulesets 26
Defining a Security Policy ............................ 26
  Accessing the Security Administration Dialog Box ........................ 27
  Description of the Security Administration Dialog Box .................. 28
Managing Tivoli NetView User Profiles ............ 29
  Description of the User Dialog Box ............ 30
  Creating and Changing a User Profile .......... 31
  Adding and Changing a Group ................... 31
  Copying a Group ................................ 33
  Deleting Group Permissions ..................... 34
  Viewing a Group’s Permissions ................ 34
  Deleting a Group ................................ 34
  Setting User Environment Variables ........... 34
Defining the Global Security Settings ............ 34
  Description of the Global Settings Dialog Box .................. 35
Steps ............................................... 37
Changing the Defaults for Interaction with NetSP 37
Managing Security .................................... 37
  Managing Logged In Users ....................... 38
  Distributing the Security Configuration ....... 38
  Reviewing Audit Data ............................. 40
Converting ARFs to SRFs ............................ 42
Verifying Security Permission For Shell Scripts 42

Chapter 3. Creating and Customizing
Submaps ............................................. 43

Objects ............................................. 43
  Displaying the Object Database ................ 43
Symbols ............................................. 43
  Defining Symbol Characteristics .............. 44
  Indicating Symbol Status ....................... 45
Maps ............................................. 48
  Information in the Map Database .............. 48
Deactivating an Event Filter ......................................... 150
Activating a Filter in a Dynamic Workspace .................. 150
Activating a Trap-to-Alert Filter (AIX Only) .................... 151
Configuring Events .................................................. 153
Advantages to Configuring Events ................................. 153
Customizing Traps .................................................... 154
Verifying Trap Customization ....................................... 156
Using the addtrap Command to Configure Events .......... 157
Displaying a Warning Window for Events ...................... 158
Example Shell Script .................................................. 158
Converting Events to Alerts (AIX Only) ......................... 159
Using the addalert command (AIX Only) ......................... 160
Sending Alerts to the Host Program (AIX Only) ............... 160

Chapter 6. Managing Network Configuration .................... 163
Discovering the Network .............................................. 163
Automatic Network Discovery ....................................... 163
Using a Seed File to Control Network Discovery ............ 165
Discovering Open Topology Networks ............................ 166
Discovering Topology Using the Openmon Application ........ 166
Discovering Cisco Routers ........................................... 167
Configuring Symbol Creation Time and Buffer ............. 167
Increasing the owdb Cache Size ................................... 168
Monitoring the Network Using Polling ......................... 169
Enabling and Disabling Polling ..................................... 169
Using SNMP for Status Polling .................................... 171
Understanding Router Fault Isolation ......................... 172
Monitoring Network Configuration ............................... 174
Listing IP Addresses for Remote SNMP Nodes ............... 174
Checking Configured Interfaces ................................... 175
Viewing Routing Table Information ............................. 175
Obtaining ARP Cache Information ............................... 175
Listing Configured Protocols ..................................... 176
Discovering Services on Nodes .................................... 176
Working With the Servmon Daemon ............................... 177
Retrieving MIB Configuration Information ..................... 180
Displaying MIB Interface Information .......................... 180
Displaying MIB System Information ............................. 180
Configuring SNMP Nodes .......................................... 181
How an Application Uses SNMP Configuration Dialog Box .... 182
Description of the SNMP Configuration Dialog Box .......... 182
Steps ........................................................................ 183
Configuring a Backup Manager .................................... 184
Configuring Manager-Container Associations ............... 185
Managing a Backup Session ....................................... 190

Chapter 7. Managing Network Performance ..................... 193
Loading and Unloading SNMP V1 MIBs ......................... 193
Loading SNMP V1 MIBs ............................................. 194
Procedure for Unloading SNMP V1 MIBs ..................... 195
Browsing SNMP V1 MIBs .......................................... 195
Using the Tivoli NetView Performance Applications .......... 197
Comparison of MIB Applications .................................. 197
Monitoring Real-Time Network Performance .................. 198
Collecting Historical Performance Information .......... 203
Monitoring File System and Paging Space .................... 209
The Monitor File System & Paging Space Dialog Box ........ 209
Starting and Stopping Monitoring Process ................... 211
Starting from the Command Line ............................... 212
Monitoring Specific Events ....................................... 212
Using the Tivoli NetView Graph Applications ............... 213
Starting a Graph Application ..................................... 213
Saving Performance Data ......................................... 213
Printing Graphed Data ............................................. 214
Generating Performance Reports ............................... 215
Contents of the Report Directory ............................... 216
Writing Reports ...................................................... 217

Chapter 8. Using the Agent Policy Manager (APM) .......... 219
What the APM Can Do for You .................................... 219
APM SmartSet Icons That You Get ............................... 220
Automatically ........................................................ 220
Propagation of Policy Status to Map Icons .................... 221
Types of APM Policies .............................................. 221
Threshold Policy ...................................................... 221
File Monitor Policy ................................................... 222
Command Policy ....................................................... 223
Filter Policy ............................................................ 223
Trap Destination Policy ............................................. 224
Analysis Policy ........................................................ 224
Alias Policy .............................................................. 224
Administration Policy ............................................... 224
Configuring and Starting the APM Daemon .................... 225
Configuring the APM Daemon through the Server Setup Application ........................................ 225
Starting the APM Daemon from the Command Line ........ 225
Starting the APM Configuration Application .................. 226
Distribution Status States ......................................... 227
Creating, Changing, and Distributing an APM Policy ...... 227
Creating or Copying an APM Policy ............................... 228
Viewing Policy Details .............................................. 229
Distributing a Policy to Remote Nodes ......................... 229
Managing Policy Distribution Using the Node Distribution Status Dialog Box .................................... 230
Modifying an Existing Policy ....................................... 231
Viewing Pending Policies .......................................... 232
Deleting an Existing APM Policy .................................. 232
Undoing a Change ..................................................... 233
Fixing an APM Policy ................................................ 233
Using the C5Maint Command ....................................... 233
Using the APMSetStatus Command ....................... 234
Copying an APM Policy from One System to Another .......... 235
Completing the APM Policy Dialog Boxes ...................... 235
Threshold and Data Collection Policy Dialog Box .................. 236
File Monitor Policy Dialog Box ................................... 236
Command Policy Dialog Box ...................................... 243
Filter Policy Dialog Box ........................................... 247
Trap Destination Policy Dialog Box ............................. 255
Analysis Policy Dialog Box ...................................... 257

Contents V
| Alias Policy Dialog Box          | 260 |
| Administration Policy Dialog Box | 262 |
| Diagnosing Problems Using the Problem Determination Assistance Facility | 264 |
| Resetting the Color of a File Monitor Icon | 266 |
| Starting Applications from the Problem Determination Assistance Dialog Box | 267 |
| Adding an Application to the Problem Determination Assistance Dialog Box | 267 |
| Example of Defining and Distributing a Policy | 269 |
| APM Reference | 272 |
| Configuring Community Names | 272 |
| APM Aliases | 274 |
| APM MLM Domains | 274 |
| Matched, Armed, and Disarmed Command Environment Variables | 275 |
| Operators for MIB Variable Expressions | 278 |
| Functions for MIB Variable Expressions | 278 |

**Appendix A. Tivoli NetView Internal Traps**

| Trap list | 283 |

**Appendix B. Tivoli NetView Events for the Tivoli Enterprise Console**

| Event Mapping and New Class Structure | 315 |
| Mappings between Events and New Class Structure | 316 |

**Appendix C. Notices**

| Trademarks | 321 |

**Appendix D. Additional Copyright and License Information**

| Index | 325 |
About This Book

This document provides a foundation for understanding and operating the IBM® Tivoli® NetView® for UNIX® program. This product is also referred to as the NetView program in this document. It explains what this product does and how to use it to manage and monitor a multiprotocol network.

Who Should Read This Guide

This document is for system administrators or anyone with a basic familiarity with the Tivoli NetView for UNIX product. You should also have a basic understanding of networking and UNIX-based systems. Most of the tasks in this book require a read-write map and root authority.

What This Guide Contains

This document contains the following information:

- **Chapter 1, “Understanding Tivoli NetView Processes”, on page 1**
  Describes the processes, applications, and databases used by the Tivoli NetView for UNIX product. Read this chapter if you need an overview of the Tivoli NetView product.

- **Chapter 2, “Defining and Managing a Security Policy”, on page 21**
  This is intended for security administrators or anyone who is responsible for managing Tivoli NetView security. This chapter describes how to use Tivoli NetView security services to control access to the Tivoli NetView product and includes information about how to create user and group profiles, collect and view audit data, and distribute a central security policy to other servers in your network.

- **Chapter 3, “Creating and Customizing Submaps”, on page 43**
  Provides information about creating a customized submap hierarchy, which includes creating and customizing maps, submaps, and objects. This chapter also includes information about how to manage maps in a distributed network environment (client/server). In addition, this chapter describes how to use the SmartSet facility to group objects together into a SmartSet. Defining a SmartSet creates a submap of objects that meet the definition criteria you specify.

- **Chapter 4, “Customizing the Graphical Interface”, on page 95**
  Explains how to customize the graphical interface. After your maps and submaps are created, use this information to customize the presentation of information. For example, you can arrange symbols, assign maps, and change the background.

- **Chapter 5, “Correlating, Filtering, and Configuring Events”, on page 109**
  Presents information about creating event correlation rules and defining event filters to control the events that are displayed. This chapter also describes how to configure events.

- **Chapter 6, “Managing Network Configuration”, on page 163**
  Describes how to manage network configuration using some of the tools and menu operations provided by the Tivoli NetView product. This chapter contains information about configuring for manager backup.

- **Chapter 7, “Managing Network Performance”, on page 193**
Describes how to monitor network performance using some of the tools and menu options provided by the Tivoli NetView product. This chapter contains information about monitoring the system resources on your management system.

- **Chapter 8, “Using the Agent Policy Manager (APM)”, on page 219**
  Describes how to use the Agent Policy Manager to set up and view information about thresholds and file monitoring in a network. You must have the Tivoli NetView Mid-Level Manager or System Information Agent installed in your network to use the Agent Policy Manager.

- **Appendix A, “Tivoli NetView Internal Traps”, on page 281**
  Lists the traps generated by the Tivoli NetView product.

- **Appendix B, “Tivoli NetView Events for the Tivoli Enterprise Console”, on page 315**
  Provides information about the events that are forwarded by the Tivoli NetView product to the Tivoli Enterprise Console® product.

---

**Publications**

This section lists publications in the Tivoli NetView for UNIX library and any other related documents. It also describes how to access Tivoli publications online and how to order Tivoli publications.

**IBM Tivoli NetView for UNIX Library**

The following documents are available in the IBM Tivoli NetView for UNIX library for Version 7.1.4:

- **Tivoli NetView Administrator’s Guide**
- **Tivoli NetView Administrator’s Reference**
- **Tivoli NetView Database Guide**
- **Tivoli NetView Host Connection**
- **Tivoli NetView Configuration Guide**
- **Tivoli NetView MLM User’s Guide**
- **Tivoli NetView Programmer’s Guide**
- **Tivoli NetView Programmer’s Reference**
- **Tivoli NetView User’s Guide for Beginners**
- **Tivoli NetView Web Console User’s Guide**

**Related publications**

The Tivoli Software Glossary includes definitions for many of the technical terms related to Tivoli software. The Tivoli Software Glossary is available, in English only, at the following Tivoli software library Web site:


Access the glossary by clicking the Glossary link on the left pane of the Tivoli software library window.

**Accessing publications online**

The product CD contains the publications that are in the product library. The format of the publications is PDF, HTML, or both. You can access the publications from the top-level directory on the product CD, or else, from the console, use the Help —> Books Online menu item.

The online help facility provides task and user interface information.
The online books are available in HTML and PDF versions (DynaText is no longer supported). The HTML versions are accessible from the Tivoli NetView Console using the Help —> Books Online menu item, which displays the books in your default web browser.

PDF versions are available in the `\usr\ov\books\C\pdf` directory. If you have installed a non-English version of Tivoli NetView, replace the C subdirectory with the appropriate locale specifier.

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli Software Information Center Web site. The Tivoli Software Information Center is located at the following Web address:


Scroll down and click the Product manuals link. In the Tivoli Technical Product Documents Alphabetical Listing window, click the `<Your Product Library Name>` link to access the product library at the Tivoli Information Center.

**Note:** If you print PDF documents on other than letter-sized paper, select the Fit to page check box in the Adobe Acrobat Print window. This option is available when you click File → Print. Fit to page ensures that the full dimensions of a letter-sized page print on the paper that you are using.

**Ordering publications**

You can order many Tivoli publications online at the following Web site:


You can also order by telephone by calling one of these numbers:

- In the United States: 800-879-2755
- In Canada: 800-426-4968

In other countries, see the following Web site for a list of telephone numbers:

http://www.ibm.com/software/tivoli/order-lit/

**Contacting software support**

If you have a problem with any Tivoli product, refer to the following IBM Software Support Web site:


If you want to contact software support, see the IBM Software Support Guide at the following Web site:

http://techsupport.services.ibm.com/guides/handbook.html

The guide provides information about how to contact IBM Software Support, depending on the severity of your problem, and the following information:

- Registration and eligibility
• Telephone numbers and e-mail addresses, depending on the country in which you are located
• Information you must have before contacting IBM Software Support

Typeface Conventions

This guide uses several typeface conventions for special terms and actions. These conventions have the following meaning:

**Bold** Commands, keywords, file names, authorization roles, URLs, or other information that you must use literally appear in **bold**. The names of titles of screen objects also appear in **bold**.

*Italics* Variables and values that you specify appear in *italics*. Words and phrases that are emphasized also appear in *italics*.

**Bold Italic** New terms appear in **bold italic** when they are defined in text.

**Monospace** Code examples, output and system messages appear in a monospace font.

Accessibility Information

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in this product enable users to:

• Operate specific or equivalent features using only the keyboard.
• Keyboard Navigation of the User Interface.

Keyboard Access

Standard shortcut and accelerator keys are used by the product and are documented by the operating system. Refer to the documentation provided by your operating system for more information.

Participating in newsgroups

User groups provide software professionals with a forum for communicating ideas, technical expertise, and experiences related to the product. They are located on the Internet and are available using standard news reader programs. These groups are primarily intended for user-to-user communication and are not a replacement for formal support.

To access a newsgroup, use the instructions appropriate for your browser.

Use these instructions for a Microsoft® Internet Explorer browser.

1. Open an Internet Explorer browser.
2. From the **Tools** menu, click **Internet Options**.
3. On the Internet Options window, click the **Programs** tab.
4. In the **Newsgroups** list, click the Down Arrow and then click **Outlook Express**.
5. Click **OK**.
6. Close your Internet Explorer browser and then open it again.
7. Cut and paste the newsgroup address of a product into the browser **Address** field, and press Enter to open the newsgroup.
Use these instructions for a Netscape Navigator browser.

1. Open a Netscape Navigator browser.
2. From the **Edit** menu, click **Preferences**. The Preferences window is displayed.
3. In the **Category** view, click **Mail & Newsgroups** to display the Mail & Newsgroups settings.
4. Select the **Use Netscape mail as the default mail application** check box.
5. Click **OK**.
6. Close your Netscape Navigator browser and then open it again.
7. Cut and paste the newsgroup address of a product into the browser **Address** field, and press Enter to open the newsgroup.

**IBM Tivoli NetView for UNIX and IBM Tivoli NetView for Windows®**

```
news://news.software.ibm.com/ibm.software.tivoli.netview-unix-windows
```

**IBM Tivoli Enterprise Console**

```
news://news.software.ibm.com/ibm.software.tivoli.enterprise-console
```
Chapter 1. Understanding Tivoli NetView Processes

The Tivoli NetView product uses many processes and databases to perform network management functions. To administer your network effectively, you should be familiar with the basic operation and interactions among the parts of the Tivoli NetView product.

This chapter describes each part of the Tivoli NetView product and describes its operation. Read this chapter if you want to understand how the different parts of the Tivoli NetView program interface.

The Tivoli NetView product uses two types of processes:

- **Foreground processes**, or applications, that run or can be invoked while the graphical interface is running.
- **Background processes**, or daemons, that run continuously regardless of whether the graphical interface is running. These processes can be started only by the root user or the root shell and stopped only by the root user. Generally, the daemons provide services that must be available at all times.

See “Using First Failure Data Capture to Resolve Daemon Problems” on page 16 for information about using First Failure Data Capture (FFDC) to help you resolve daemon problems.

### Foreground Processes

The processes, or applications, listed in Table 1 provide the interface through which you manage your network. Many of these processes correspond to selections you can make from the main menu.

<table>
<thead>
<tr>
<th>Process Category</th>
<th>Process Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Principal Tivoli NetView graphical interface</td>
<td>ovw</td>
</tr>
<tr>
<td>Map display</td>
<td>ipmap (IP topology)</td>
</tr>
<tr>
<td></td>
<td>xxmap (non-IP topology)</td>
</tr>
<tr>
<td>Event display</td>
<td>nvevents</td>
</tr>
<tr>
<td>Menu operations</td>
<td>xnmtrap</td>
</tr>
<tr>
<td></td>
<td>xnmnsnmpconf</td>
</tr>
<tr>
<td></td>
<td>xnmloadmib</td>
</tr>
<tr>
<td></td>
<td>xnmbrowser</td>
</tr>
<tr>
<td></td>
<td>xnmcollect</td>
</tr>
<tr>
<td></td>
<td>xnmbuilder</td>
</tr>
<tr>
<td></td>
<td>xnmfault</td>
</tr>
<tr>
<td></td>
<td>xnmgraph</td>
</tr>
<tr>
<td></td>
<td>xnmrunreport</td>
</tr>
<tr>
<td></td>
<td>nmpolling</td>
</tr>
<tr>
<td></td>
<td>backup</td>
</tr>
<tr>
<td></td>
<td>shpmon</td>
</tr>
<tr>
<td>Dialog box management</td>
<td>xnmappmon</td>
</tr>
</tbody>
</table>
ovw Application

The /usr/OV/bin/ovw application is the principal Tivoli NetView graphical interface for managing TCP/IP-based internets. The ovw application provides map drawing, map editing, and menu management operations. The ovw application is an X11/Motif application based on OSF/Motif user interface guidelines.

To start the ovw application, execute the /usr/OV/bin/netview shell script. The netview shell script supports the same options as those supported by the ovw application. The ovw application automatically starts the ipmap, xxmap, and nvevents applications.

If you are the root user, the netview shell script starts the daemon processes and the ovw application. If you are not the root user, the shell script starts only the ovw application.

After the ovw application is operating, some daemons and applications are dynamically updated through events forwarded by the ovesmd daemon. Table 2 shows where the ovesmd daemon sends events, and how those events are used.

Table 2. Where the ovesmd Daemon Sends Events

<table>
<thead>
<tr>
<th>Destination</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>The netmon daemon</td>
<td>Tracks changes in the state of the network.</td>
</tr>
<tr>
<td>The ovtopmd daemon</td>
<td>Updates the topology database.</td>
</tr>
<tr>
<td>The ipmap application</td>
<td>Informs the graphical interface of changes in the IP topology database and informs ovtopmd of user-initiated changes to map databases.</td>
</tr>
<tr>
<td>The nvevents application</td>
<td>Displays the event in either the Event Cards or List format.</td>
</tr>
<tr>
<td>The tralertd daemon (AIX® only)</td>
<td>Forwards events to the Tivoli NetView for z/OS product as alerts.</td>
</tr>
</tbody>
</table>

ipmap Application

The /usr/OV/bin/ipmap application is started by the ovw application automatically. The ipmap application ensures that the ovw application (graphical interface) and the ovtopmd daemon behave consistently. For example, when an object is deleted using the graphical interface, the graphical interface tells ipmap which symbols and objects were removed. The ipmap application then tells ovtopmd to make the appropriate changes to the topology database.

In another example, when the netmon daemon discovers a new node, the ovtopmd daemon adds the node to the IP topology database and informs ipmap that a new node has been discovered. The ipmap application uses what it knows about IP devices to tell the graphical interface which icon and connection symbols it needs to create. The graphical interface then displays the correct symbols and modifies the map database accordingly.

When the ipmap application is started, it queries the ovw application to determine when the map was last open. Next, it calls the ovtopmd daemon to find out all changes to the IP topology database since the last time the map was open using the graphical interface. The ipmap application determines what, if anything, has
changed since the map was last open, and then tells the ovw application to add, change, or delete the appropriate icon and connection symbols. This process is called synchronization.

If the ipmap application can find an association between an SNMP node’s sysObjectID Management Information Base (MIB) variable and a symbol type in the oid_to_sym file, the graphical interface displays the node by drawing the appropriate symbol on the submap. If the oid_to_sym file does not contain a matching sysObjectID entry, the ipmap application extracts attributes from the topology database and tells the ovw application to create and display a generic symbol.

After the synchronization phase is completed, the ipmap application is updated based on information received dynamically from the ovtopmd daemon, which receives updates from the netmon daemon. The netmon daemon continuously monitors the state of your network by sending SNMP requests to SNMP-managed nodes and Internet Control Message Protocol (ICMP) requests to non-SNMP IP nodes. The netmon daemon communicates changes to the ovtopmd daemon, which updates the topology database and informs the ovwdb daemon that the object database needs to be updated. If the change affects the IP Map, the ipmap application notifies the ovw application. Otherwise, the change is reflected only in the object database and the IP topology database.

xxmap Application

The /usr/OV/bin/xxmap application is very similar to the ipmap application with one important difference. The xxmap application processes non-IP topology information that is stored by the gtmd daemon in the general topology database. The ipmap application processes IP information that is stored by the ovtopmd daemon in the IP topology database. The non-IP topology information stored in the general topology database comes from non-IP discovery applications or agents that use the general topology MIB format to send topology information to the gtmd daemon.

The xxmap application ensures that the maps being displayed are synchronized with the contents of both the general topology and object databases. Using the /usr/OV/conf/oid_to_protocol file, the xxmap application matches the oid contained in the MIB to the correct submaps and symbols required to display the topology of a particular protocol. For more information about the xxmap application, refer to Tivoli NetView for UNIX Programmer’s Guide.

nvevents Application

The /usr/OV/bin/nvevents application displays events in the main window Control Desk in either the Event Cards or List presentation format. When this process is invoked by the ovw application, it reads the ovevent.log file to recover events that occurred since the nvevents application was last running.

Following startup, nvevents receives SNMP traps that are filtered by the ovesmd daemon and uses those traps to update the event cards or list.

It also monitors the status of the snmpCollect, netmon, or ovtopmd daemons. If one of those daemons is not running, nvevents presents a warning box to notify you.
xnmtrap Application

The /usr/OV/bin/xnmtrap application is invoked when you click Event Configuration —> Trap Customization: SNMP in the Options pull-down menu. This option helps you control how enterprise-specific events (traps) are handled. For example, you can customize the message displayed through nvevents when a particular event arrives. You can also specify a command or a script that should be executed when a particular event arrives. Event configuration changes are stored in the /usr/OV/conf/C_trapd.conf configuration file.

xnmsnmpconf Application

The /usr/OV/bin/xnmsnmpconf application is invoked when you click SNMP Configuration in the Options pull-down menu, which enables you to specify netmon status polling intervals, timeout intervals, number of retries, proxy information, and agent community names.

The SNMP configuration values are stored in the /usr/OV/conf/ovsnmp.conf_db file. Processes on the management station, such as the netmon daemon, MIB applications, and SNMP API-based applications, look up an agent’s community name and other SNMP options in the ovsnmp.conf_db file when access to agent MIB values through SNMP requests is required.

xnmloadmib Application

The /usr/OV/bin/xnmloadmib application is invoked when you click Load/Unload SNMP V1 MIBs in the Options pull-down menu. Click this option when you want to load new SNMPv1 Internet-standard, enterprise-specific, or general topology MIBs into the MIB database.

After you have loaded the new MIB into the MIB database, you can use the MIB Browser, MIB Data Collection, and MIB Application Builder options, as well as the applications built by the MIB application builder, to manage your multivendor network.

Web Console MIB Loader Application

The SNMP V1/V2 Web Console MIB Loader is invoked when you click Tools —> Web Console MIB Loader pull-down menu. Select this option when you want to load new SNMPv1 or SNMPv2 Internet-standard, enterprise-specific, or general topology MIBs for the Web Console MIB Browser. The initial MIB Loader window displays a list of the loaded MIBs. See the Tivoli NetView Web Console User’s Guide and Tivoli NetView Release Notes, Version 7.1 for more information.

Web Console MIB Browser Application


xnmbrowser Application

The /usr/OV/bin/xnmbrowser application is invoked when you click MIB Browser SNMP V1 in the Tools pull-down menu, which enables you to get and set MIB values for Internet-standard and enterprise-specific MIB objects on SNMPv1 agents. This application hides the actual SNMP requests used to perform these operations, so you need to know only which MIB objects you want to access, not the commands required to do so.
xnmcollect Application

The `/usr/OV/bin/xnmcollect` application is invoked when you click **Data Collection and Thresholds: SNMP** in the **Tools** pull-down menu, which enables you to configure the manager to collect MIB data from network objects at regular intervals. The configuration information is stored in the `/usr/OV/conf/snmpCol.conf` configuration file. The collected data is stored in files in the `/usr/OV/databases/snmpCollect` directory.

After you have stored the collected data, you can use the xnmgraph application to view the collected data or import the data into your own application using the `snmpColDump` command. You can also define thresholds for the collected MIB data and generate events when the specified thresholds are exceeded.

xnmbuilder Application

The `/usr/OV/bin/xnmbuilder` application is invoked when you click **MIB Application Builder: SNMP** in the **Tools** pull-down menu, which enables you to build custom screens to manage multivendor MIB objects. The information you define using the MIB application builder is stored in registration files and help files. The xnmgraph application displays Graph applications, while the xnmappmon application displays Form and Table applications.

xnmgraph Application

The `/usr/OV/bin/xnmgraph` application enables you to graph the results of monitoring operations for managed SNMP objects selected from the map. The results might be real time or collected historical data.

When you click **Graph Collected Data: SNMP** in the **Tools** pull-down menu, the owv application forwards the objects previously selected on the map to the xnmgraph application. The xnmgraph application queries the selected objects and displays the results in a line graph.

xnmfault Application

The `/usr/OV/bin/xnmfault` application enables you to locate, for any node on the map, all of its component resources that have failed.

When you select a node on a submap, and then select **Tools —> Failing Resource Display** the xnmfault application creates a submap showing all the resources that are children of the selected node and that have failed. Be sure to close this submap through the navigation tree. If not, the xnmfault application is not notified that the submap has been closed. The submap remains open and can be accessed from the Navigation Tree.

xnmrunreport Application

The `/usr/OV/bin/xnmrunreport` application corresponds to the **Reports: Site Provided** option from the **Monitor** pull-down menu. You can use this application to generate reports that send output directly to the screen, which is useful for problem determination or real-time network monitoring. You can also generate reports that store information in flat files or databases for future reference.

nmpolling Application

The `/usr/OV/bin/nmpolling` application is accessed through the **Topology/Status Polling Intervals: IP** option from the **Options** pull-down menu that enables you to configure certain netmon polling intervals.
backup Application
The /usr/OV/bin/backup application is accessed through the Backup option from the Administer pull-down menu. This process lets you configure managers to backup another manager’s objects if the other manager should go down. You configure objects as manager nodes or managed containers and then specify managers as an active manager or a backup manager.

shpmon Application
The /usr/OV/bin/shpmon application is accessed through the Local Filesystem & Paging Space option from the Monitor pull-down menu. Using this option, you can receive a trap or message letting you know when a threshold condition has reached its limit. The shpmon application monitors the root file system and the paging space on the local manager system where the Tivoli NetView product is installed.

xnmappmon Application
The /usr/OV/bin/xnmappmon application manages dialog boxes that contain the text output of monitoring operations performed on managed SNMP objects that have been selected from the map. This application is also called an application encapsulator.

When you click a monitoring operation in the Tivoli NetView main menu bar, the ovw application forwards the objects selected on the map to the xnmappmon application as input. The xnmappmon application displays the appropriate dialog box, translates the selected operation and object or objects into a command, and executes the command.

When processing is completed, the command returns its text output to the xnmappmon application for display in the dialog box. In the case of the Locate—>Route menu option, xnmappmon also returns to ovw a list of map objects to highlight as output of the network management operation.

For example, the xnmappmon application can be used to execute the mibform, mibtable, rnetstat, findroute, rbdf, and rping commands. These commands help you monitor and diagnose problems in your TCP/IP network.

Web-Based Device Management
You can use Web-based home pages to manage objects and devices.

Some network objects can use Web-based home pages for configuration and monitoring. The Tivoli NetView console supports this capability, enabling access from either the shortcut menus for the object (Management Page) or the main menu (Tools -> Web Device Mgmt -> Management Page). This feature is also available from the Web Client. To enable Web-based home pages, modify the file /usr/OV/conf/oid_to_type; identify the sysObjectIds of the devices that support Web-based management and add the W flag to those lines as per the instructions in the heading of the file. The management home page for a node is assumed to be http://hostname. To use a different URL, modify the ManagementURL field in the HTTP section of the Object Description dialog.

Accessing the NetView Web Client Home Page
To access the Tivoli NetView Web Client home page for a Tivoli NetView server from the Tivoli NetView Console, select the node on the submap and click Tools.
Process Management

As a network administrator, you use several process management commands to control the operation of Tivoli NetView daemons. These commands are:

- **ovstart**: Starts all daemons or selected daemons, depending on the options you specify.
- **ovstop**: Stops all daemons or selected daemons, depending on the options you specify.
- **ovstatus**: Shows the status of all daemons or selected daemons, depending on the options you specify, including the process management daemon itself.
- **nvstatus**: Shows the status of all daemons or selected daemons, depending on the options you specify, that are running on the server. This command can be run only on a client workstation.

The **ovspmd Daemon**

The process management daemon, ovspmd, coordinates the starting and stopping of daemons that communicate with the Tivoli NetView product.

The **ovstart** command starts the ovspmd daemon, which in turn starts the other Tivoli NetView daemons in a particular order. The information about startup order is in the startup configuration file, `/usr/OV/conf/ovsuf`.

The startup configuration file is constructed by the **ovaddobj** command. The ovaddobj process takes information from the local registration files (LRF) and places it in the startup configuration file.

The ovspmd daemon receives requests from the **ovstart** command and sends status responses to ovstart. It starts all Tivoli NetView daemons (if they are listed in the ovsuf file) and maintains a communication channel with each of them. These daemons should always be running. On the AIX operating system, the ovspmd daemon starts the host connection daemons, spappld and tralertd, only if the AIX Service Point is installed. A check for the AIX Service Point is performed during installation of the Tivoli NetView product.

Background Processes

The daemons fall into the following categories:

- **Topology discovery and database operation**
- **Event and trap processing**
- **Security services**
- **Service discovery and monitoring**
- **Host connection** (AIX only)

### Topology Discovery and Database Daemons

The Tivoli NetView product discovers and updates the topology of IP networks and translates the information into symbols that appear on the views you see of...
your network’s map. The program also facilitates the discovery of networks that use non-IP protocols for communication, so you can extend your network management coverage.

Figure 1 illustrates the interactions among the topology discovery and database daemons. Each daemon is described in the following sections.

**mgragentd Daemon**
The `/usr/OV/bin/mgragentd` daemon runs on the manager station to determine the status of the Tivoli NetView product and to respond to queries from other manager stations. In a distributed network environment, the mgragentd daemon is used to determine the status of the Tivoli NetView daemons. This daemon requires no user configuration or manipulation.

**netmon Daemon**
The `/usr/OV/bin/netmon` daemon polls SNMP agents to discover network topology. During initial discovery, the netmon daemon sends an SNMP trap to the trapd daemon to inform it of newly discovered network entities. Some of these entities might be agents that communicate with non-IP networks or devices. As long as the entities have an IP address, the netmon daemon will discover them.

After initial discovery is complete, the netmon daemon polls the SNMP agents to detect topology, configuration, and status changes in the IP network, and sends corresponding traps to the trapd daemon. However, the netmon daemon does not have a connection with non-IP networks or devices. Topology changes for these network entities must be communicated to the Tivoli NetView product in other ways. See "noniptopod Daemon" on page 9 and "gtmd Daemon" on page 10 for more information about non-IP topology discovery.

In addition to polling SNMP agents, the netmon daemon polls network nodes. To check an SNMP-managed node’s MIB values, the netmon daemon uses an SNMP request. To check the status of all nodes, the netmon daemon uses ICMP echo requests (ping) or SNMP queries for configured nodes. Based on the discovered information, the netmon daemon generates and updates the topology map.

The netmon polling values and information about network objects, including their relationships, status, and thresholds, are stored in a set of files called the topology database. If the database does not initially exist, the ovtopmd daemon creates it during discovery of the network’s topology and automatic generation of the map.
When you start up the graphical interface, the ipmap application compares the contents of the topology database with the contents of the graphical interface’s map databases. The ipmap application tells the graphical interface what has changed since its last invocation, and then the graphical interface updates the map.

The netmon daemon assumes that your initial network management region is composed of the network or networks to which the Tivoli NetView product, the management station, is directly connected. If you want a different initial configuration, you can provide a seed file, which contains a list of nodes you want to appear on the automatically-generated network map.

See “Using a Seed File to Control Network Discovery” on page 165 for information about seed files.

**nvlockd Daemon**

The /usr/OV/bin/nvlockd helps the gtmd daemon and xxmap application control access to the ovwdb daemon. This daemon requires no user configuration or manipulation.

**ovtopmd Daemon**

The /usr/OV/bin/ovtopmd daemon maintains the network topology database. The topology database is a set of files in the /usr/OV/databases/openview/topo directory that store netmon polling values and other information about network objects, including their relationships and status.

The ovtopmd daemon generates and updates the topology database using status information obtained from the netmon daemon. The ovtopmd daemon also checks for existing non-IP objects with which to correlate.

**ovwdb Daemon**

The /usr/OV/bin/ovwdb daemon controls the Tivoli NetView object database. This database stores object information that the graphical interface uses to generate output for Describe operations. For example, when you click **Modify/Describe —> Objects** in the **Edit** pulldown menu, you can view several attributes for the objects you have selected. The information you see is retrieved from the Tivoli NetView object database.

If the netmon daemon detects a change in the network, it calls the ovtopmd daemon to update the topology database. In turn, the ovtopmd daemon calls the ovwdb daemon to update the Tivoli NetView object database.

To run the ovw application, first start the ovwdb daemon. The object database must be accessible to the ovw application so that the default submap can be generated.

**nvcold Daemon**

The /usr/OV/bin/nvcold daemon maintains SmartSets as they have been defined by users or applications.

**noniptopod Daemon**

Initial topology discovery is handled by the netmon daemon, which sends an SNMP trap to the trapd daemon when a new network object is discovered. The trapd daemon then forwards traps, via the ovesmd daemon, to the /usr/OV/bin/noniptopod daemon, which accepts any trap that meets the following criteria:

- The discovered object has an IP address.
• The discovered object could be an agent that can communicate with a non-IP device or network.

The noniptopod daemon sends an SNMP get command for each object identifier (OID) listed in the /usr/OV/conf/oid_to_command file to the new network object associated with the trap. It checks to see if the OIDs are supported by the agent. If a non-null response is sent back, the commands associated with the supported OIDs are started.

The command in the oid_to_command file activates the non-IP protocol’s proprietary daemon, which sends a request to the agent to gather all non-IP topology information and forward it to the proprietary daemon for conversion to the general topology MIB format. The MIB information is then sent to the gtmd daemon.

gtmd Daemon
The /usr/OV/bin/gtmd daemon receives information sent by non-IP discovery applications, agents, and proxy agents that use the general topology MIB format to describe the attributes of devices on a non-IP network. Information can be received in traps or in application programming interface (API) calls. This daemon stores the non-IP topology information in its own database and correlates it with IP topology information stored in the object database to determine whether an IP object can also be identified as having an association with a non-IP protocol. The gtmd daemon makes the topology information available for display through the xxmap application.

Non-IP discovery applications can register with the gtmd daemon to receive notifications of changes to the topology information or to receive topology data. The gtmd daemon updates its database each time a trap is received and notifies registered applications of the operation performed. The gtmd and noniptopod daemons are, by default, not started when the Tivoli NetView product is started. You can use the Tivoli desktop to configure them and indicate that they should be started by the netview command.

For more information about the Tivoli NetView general topology function and the gtm API, refer to the Tivoli NetView for UNIX Programmer’s Guide.

otmd Daemon
The /usr/OV/bin/otmd daemon receives and processes dynamic topology and status updates from an openmon topology agent.

The otmd daemon reads the configuration file, in the /usr/OV/conf/openmon subdirectory, for each openmon topology application. The otmd daemon uses the ovstart command to start any openmon topology application that is configured to start automatically by the otmd daemon.

It obtains the port number for the otmd service from the /etc/services file and sets up the TCP socket and waits for the openmon agents to connect. When a connection is received from an agent, the otmd daemon receives and processes all the data from the agent. It stores all topology object information in the general topology manager (GTM) database and the attribute information with the topology object in the OVW object database.
If an update sequence number is received from an agent, the otmd daemon saves the update sequence number in the agent’s configuration file. This sequence number is maintained to keep track of the information that the agent has sent to the openmon application.

nvotd Daemon
The `/usr/OV/bin/nvotd` daemon receives non-IP topology events from the gtmd daemon and forwards them to the Event Display application (`nvevents`) if API calls are used to send information to the gtmd daemon. The nvotd daemon, by default, is not started when the Tivoli NetView product is started. You can use the Tivoli desktop to configure the nvotd daemon and indicate that the daemon should be started by the `netview` command.

C5d Daemon
The `/usr/OV/bin/C5d` daemon coordinates setting threshold monitor and file monitor definitions for Tivoli NetView Mid-Level Managers (MLMs). The C5d daemon is, by default, not started when the Tivoli NetView product is started. You can use the Server Setup application to configure the C5d daemon so that it starts when the `netview` command is used.

Event and Trap Processing Daemons
Events and traps provide information about changes in the status of network elements and alert the Tivoli NetView product to occurrences in the network. When events and traps are received, they must be routed to the appropriate applications and logged for future reference.

The daemons that perform these tasks can be divided into the following three groups:

- Communications infrastructure daemons, including the `pmd` daemon and the `orsd` daemon, that control all CMOT and some SNMP communications entering and leaving the Tivoli NetView product
- Event management services daemons, including the `ovesmd` daemon and the `ovelmd` daemon, which route and log all events received by the Tivoli NetView product
- Other event and trap processing daemons, including the `trapd` daemon, the `nvcorr` daemon, the `actionsvr` daemon, the `nvpagerd` daemon, the `nvserverd` daemon, the `trapgend` daemon (AIX only), and the `snmpCollect` daemon

Figure 2 illustrates the interactions among the event and trap processing daemons. Each daemon is described in the following sections.
pmd Daemon

When an agent sends an event to an application, the event contains no routing information. It is similar to sending a letter without an address. The /usr/OV/bin/pmd daemon receives the events from the trapd daemon and forwards them to the ovesmd daemon. The ovesmd daemon forwards the events to the applications that have registered to receive them.

The pmd daemon centralizes the external communications for all applications and processes that use the CMOT protocol, as well as some of those using SNMP. This daemon contains the SNMP and CMOT protocol stacks.

The pmd daemon has two components:

- The locator function, which routes outbound requests to the appropriate agent. The locator function consults the data in the object registration database, which includes agent locations and the protocol used to access them. Note that application programs do not access the object registration database directly; instead, the pmd daemon automatically does so on their behalf.
- The association management function, which provides a way of sharing connections between network management nodes. This function enables application managers to share connections without having to bring up multiple connections between the nodes.

orsd Daemon

The /usr/OV/bin/orsd daemon maintains the consistency of the CMIS object information in the object registration database. The object registration database is consulted by the pmd daemon to determine where an agent resides and which protocol to use to communicate with it.
ovesmd Daemon
The /usr/ov/bin/ovesmd daemon is part of Tivoli NetView event management services. Known as the event sieve agent (ESA), the ovesmd daemon distributes events throughout the network based on the filters in effect for a particular application or user.

When events are sent from agents, they do not contain any routing information that the pmd daemon can use to forward them to the correct application. When the pmd daemon receives an event from trapd, it sends the event to the ovesmd daemon, which filters the event and forwards it to applications that have registered to receive it. The ovesmd daemon also forwards the event to the ovelmd daemon, which controls the event log.

ovelmd Daemon
The /usr/ov/bin/ovelmd daemon is the event log agent (ELA), which stores SNMP traps, CMIS events, and event log configuration values in the /usr/ov/log/ovevent.log file. The ovevent.log and the ovevent.BAK files are binary files that are the source of information for the dynamic and historical event displays. It logs all SNMP traps received from the network and internal processes.

trapd Daemon
The usr/ov/bin/trapd daemon receives traps from agents and internal processes and forwards them to the netmon (AIX only), nvcorrd, and the pmd daemons, and to the ipmap application. In return, the netmon daemon sends events to trapd when its polling reveals a change in the status of a network element.

The trapd daemon also forwards network events to other applications that have connected to it through the OVsnpmp API.

The trapd daemon logs all received traps in the /usr/ov/log/trapd.log file. You can use any editor to look at the contents of the trapd.log file. You can also choose to turn trapd logging off by clicking Server Setup —> Configure —> Set options for daemons —> Set options for event and trap processing daemons —> Set options for trapd daemon.

nvcorrd Daemon
The /usr/ov/bin/nvcorrd daemon receives events from the trapd daemon, correlates or compares the events to event processing decision and actions defined in rulesets registered with nvcorrd, and forwards them to registered applications, one of which is the Event Display application. The nvcorrd daemon passes events to the actionsvr daemon to manage actions defined in event correlation rulesets.

actionsvr Daemon
When an action is to be processed in an event correlation rule, the nvcorrd daemon passes the action to the /usr/ov/bin/actionsvr daemon. The actionsvr daemon manages the action, starting a child process, while the nvcorrd daemon continues to process the event correlation ruleset. All actions requested and the events which caused those actions are logged in the /usr/ov/log/nvaction.alog and /usr/ov/log/nvaction.blog files.

nvpagerd Daemon
The /usr/ov/bin/nvpagerd daemon manages the routing of the page command that is issued from the command line or within an event correlation rule.

When the paging action is to be processed in an event correlation rule, the nvcorrd daemon passes the action to the actionsvr daemon, which passes the action to the
nvpagerd daemon. All paging actions requested and the events which caused those actions are logged in the /usr/OV/log/pagerd.log file.

**nvserverd Daemon**
The /usr/OV/bin/nvserverd daemon receives events from the nvcorrd daemon and forwards them to different Event Display applications running on client workstations. The nvserverd daemon enables you to select a set of events in one client application and work with it in all registered applications, clearing these events or changing the status of the events.

The nvserverd daemon can be configured to forward events to the Tivoli Enterprise Console. There are two versions of the nvserverd daemon:

- nvserverd sends events to a Tivoli Enterprise Console server by communicating directly with the server.
- tme_nvserverd sends events to a Tivoli Enterprise Console server using the Tivoli Management Environment® transport, which is more secure than direct communication.

Use the Server Setup function to change the version of nvserverd that is used.

**trapgend Daemon (AIX Only)**
The /usr/OV/bin/trapgend daemon is a subagent (SMUX peer) provided with the Tivoli NetView product that converts AIX alertable errors to SNMP traps.

On RS/6000 processors running on AIX, system errors are logged by the operating system’s error logging facilities in the /dev/error special file. An object installed by the Tivoli NetView product in each system’s object data manager (ODM) directs the AIX error logging daemon (errdemon) to inform the trap-notify process when alertable errors are logged. These alertable errors are forwarded by the trap-notify process to the trapgend daemon, which converts these alertable errors to SNMP traps.

Using the SNMP multiplexer (SMUX) protocol, trapgend forwards the traps to the AIX SNMP daemon, snmpd. The snmpd daemon then forwards the traps to the trapd daemon on the Tivoli NetView manager specified by the trap destination.

The trapgend daemon also provides a trap throttle to suppress identical trap generation, enables remote ping operations from the Tivoli NetView product, and supports CPU utilization and disk space monitoring MIB extensions.

The trapgend daemon must be installed on all nodes running on AIX in your network if you want to receive information about CPU and disk space utilization.

**snmpCollect Daemon**
The /usr/OV/bin/snmpCollect daemon collects, compares, and stores SNMP agent MIB values. It also checks the collected values against user-defined thresholds and generates events if the thresholds are exceeded.

The snmpCollect daemon also writes performance information to the DB2® product. This information is used by the Tivoli NetView Warehouse enablement pack.
Security Services Daemons

The security services daemons determine whether security is on or off, manage authentication and identification of Tivoli NetView users, and manage communication between the security server and client workstations. Each daemon is described in the following sections.

nvsecd Daemon

The /usr/ov/bin/nvsecd daemon determines if Tivoli NetView security is on or off. If security is on, the nvsecd daemon requires each Tivoli NetView user to login using a valid Tivoli NetView user ID, password, and group ID. The nvsecd daemon checks the user’s profile to verify login and checks the permissions defined in the user’s group profile to control access to Tivoli NetView resources. After login verification, the nvsecd daemon establishes a shared-key security context for each Tivoli NetView process. The nvsecd daemon also monitors and stores security audit data.

nvsecltd Daemon

The /usr/ov/bin/nvsecltd daemon runs on client workstations and listens for communication from the security server. The nvsecltd daemon starts only when security is turned on and when at least one user logs into the Tivoli NetView product on a client workstation. The nvsecltd daemon stops when the last client user logs out of the Tivoli NetView product.

Service Discovery and Monitoring Daemon

Part of managing a network involves knowing which nodes provide important services, such as DB2 servers and WebSphere® Application servers, and monitoring the status of these services. The service monitor application provides the ability to discover and monitor these services.

servmon Daemon

The service monitor application is implemented as the servmon daemon. The servmon daemon discovers and monitors services on Tivoli NetView managed nodes. The servmon daemon uses a default configuration file to specify which services to discover, the discovery and monitoring methods, and how often to check the status of all discovered services. See “Discovering Services on Nodes” on page 176 for more information.

Service objects are created in the object database for each service that is discovered on a node when a Service SmartSet is specified in the respective configuration file entry. Status monitoring is not performed either for entries that do not specify a Service SmartSet, or for entries that specify a value of 0 (zero) for the status interval field.

Service objects contain status. The service status does not contribute to the overall IP status of a node.

When a service is discovered, the servmon daemon creates a Boolean field on the node object. The field value is set to true, and the node automatically becomes a member of the designated Service SmartSet.

If a service is removed from a node, servmon removes the designated Boolean field from the node after it has determined that the service has been unavailable for a specified amount of time.
Host Connection Daemons (AIX Only)

If your Tivoli NetView product is connected to the Tivoli NetView for z/OS product by an AIX Service Point program, you are using the services of the host connection daemons. These daemons facilitate both the conversion of SNMP traps to SNA alerts as well as all communication between the Tivoli NetView product and the Tivoli NetView for z/OS product through the Service Point application. Each daemon is described in the following sections. For more information about the host connection, refer to **Tivoli NetView for UNIX Host Connection**.

**tralertd Daemon (AIX Only)**

The `/usr/OV/bin/tralertd` daemon is used in an environment where both TCP/IP and SNA protocols are running. The `tralertd` daemon receives events and traps generated or received by the Tivoli NetView product. If a trap is so configured, the `tralertd` daemon converts it to an SNA alert, and sends the SNA alert to the Tivoli NetView for z/OS product through the AIX NetView Service Point product. If all the converted trap information cannot fit into the SNA alert, the original trap information is saved in the `/usr/OV/databases/tralertd` database and assigned a corresponding Log ID. The Tivoli NetView for z/OS product uses this Log ID to query the `tralertd` database to retrieve the rest of the information.

**spappld Daemon (AIX Only)**

The `/usr/OV/bin/spappld` daemon provides a command interface between the Tivoli NetView for z/OS product on the host and the Tivoli NetView product in an environment running SNA and TCP/IP protocols. The `spappld` daemon receives Tivoli NetView for z/OS RUNCMDs and runs their contents in the internet environment, and sends responses to the Tivoli NetView for z/OS product through the AIX NetView Service Point product.

---

**Using First Failure Data Capture to Resolve Daemon Problems**

The first failure data capture (FFDC) function provides a means to collect data after the first failure of a Tivoli NetView daemon. The FFDC function automatically collects and archives necessary data at the time of failure. You can use this information to analyze the problem, or you can send the data archive to IBM customer support.

**Working With the FFDC Function**

Use the `/usr/OV/conf/FFDC/FFDC.properties` file to control and configure the FFDC function. This file must have run permission, because it is called from Korn shell (ksh) scripts.

The scripts are started by the `ovspmd` daemon whenever a daemon abends (for example, a startup failure or a core dump). Note that the scripts are not started if a daemon is stopped by a user with the `ovstop` command. These scripts are only started for registered daemons that are shipped with NetView 7.1.4. The scripts are located in `/usr/OV/conf/FFDC/scripts`. Unless you are directed to do so by IBM customer support, do not modify the files in this directory, because unpredictable results can occur.

Table 3 on page 17 provides a description of the `FFDC.properties` file statements.

**Notes:**

1. You do not have to restart the Tivoli NetView product after you change a value, because FFDC data collection is initiated by the ksh scripts.
2. Default values are provided.
### FFDC Storage Information

FFDC data files are stored in the NV_FFDC_DIR directory. The files are stored in subdirectories that are named using the date that the data was collected. Temporary space that is used during the archiving process is also defined in this directory.

The amount of storage that is defined must be at least twice the sum of the largest possible files that are collected. Note that this value might need to be increased if you have other types of tracing enabled. When the NV_FFDC_DIR directory is filled to capacity, the oldest data archives are deleted, and the latest information can be stored.

### Databases

This section describes the following databases:

- Map
- Object
- IP topology
- General
- Object registration service

These databases are controlled by the Tivoli NetView processes; you cannot edit them directly.

#### Map Database

The map database contains presentation information that is specific to each map. There is one map database per map. Examples of presentation information stored in the map database include the exact symbol placement on a map, the symbol associated with each object, and symbol labels. The ovw application updates the map database based on requests from the user or from other processes, such as the ipmap application. Use the ovmapdump command to view the contents of the map database. The map database is maintained by the ovw application.

#### Object Database

The object database contains global object information. The information is generic; that is, it is not customized to any specific application. The object database contains information related to fields such as sysObjectID, vendor, and SNMP agent. When you choose the Edit -> Modify/Describe Object menu option, the information you see in the fields comes from this database. If netmon detects a status change in the network, netmon calls ovtopmd which calls ovwdb to update the object.
database. Use the `ovobjprint` command to view the object database. The object database is maintained by the ovwdb daemon.

**IP Topology Database**

The IP topology database contains topology information used during IP discovery and layout. The information in the topology database spans all maps. Much of the information in the Tivoli NetView object database is duplicated in the topology database. Information in the topology database that is not in the object database includes state information for the netmon daemon.

The most important netmon state information includes time stamps that indicate when the object last changed and should next be polled. This information helps the netmon daemon detect changes so it can communicate the changes to the ovtopmd and trapd daemons. The topology database is controlled by the ovtopmd daemon and is updated based on information received from the netmon daemon. Use the `ovtopodump` command to view the contents of the IP topology database. The IP topology database is maintained by the ovtopmd daemon.

**General Topology Manager Database**

The general topology manager (GTM) database stores topology information sent to the gtmd daemon. This database also stores information about submap grouping and content that has been defined by the protocol discovery applications or agents.

The `xxmap` application queries both this database and the Tivoli NetView object database for display and semantic information. Any topology information stored in the general topology database can be deleted only by the agent that originally added the information. You cannot view the GTM database.

For more information about the use of the GTM database refer to *Tivoli NetView for UNIX Programmer’s Guide.*

**Object Registration Service Database**

The object registration service (ORS) database contains location and protocol information for agents that use a protocol other than SNMP. This information helps to provide location transparency, which allows managers and agents to access objects and agents without using hard-coded addresses.

**Customizing and Generating Reports on the Object Database**

This section describes two utilities that allow you to customize and query the object database from the command line. Using these utilities substantially reduces the effort to customize the object database and produce reports. They also allow integration with Tivoli Inventory to collect and retain hardware and software information across the enterprise.

**The nvdbimport Utility**

This nvdbimport utility can be used to import object field values into the Tivoli NetView object database from a file. For example, if you wanted to set a new field, called `isSite2`, to TRUE for all nodes at Site 2, you would perform the following steps:

1. Create an input file with the list of nodes that are at Site2.
2. Add the field `isSite2` with the value of TRUE.
3. Create a field called `isSite2` in the object database.
4. Run the nvdbimport utility to update the database for these nodes.

You can also create a SmartSet of all nodes with isSite2 set to TRUE.

If you need to recreate the database, you can quickly restore your customization. See the *Tivoli NetView Administrator’s Reference*, for more information. Example input files (called *.import) are provided in the /usr/OV/conf/nvdbtools directory.

**The nvdbformat Utility**

The nvdbformat utility can be used to generate custom reports from the Tivoli NetView object database complete with header and footers that provide aggregate information. For more information, see the *Tivoli NetView Administrator’s Reference*. Example input files (called *.format) are provided in the /usr/OV/conf/nvdbtools directory.

---

**Role of the SEAs**

This section applies only to the Solaris Operating Environment (hereinafter referred to as Solaris).

The two Sun Solstice Enterprise Agents (SEAs) relevant to the Tivoli NetView product are snmpdx and mibiisa.

The snmpdx agent is the master agent that controls the other agents. It also routes incoming SNMP requests similar to the way snmpd works on AIX. In summary, this agent listens on Port 161 and routes the requests to the corresponding SNMP agent based on the registration files residing in the /etc/snmp/conf directory.

The mibiisa agent provides support for MIB-II.

See the *Tivoli NetView Release Notes, Version 7.1* for information about how to install the documentation and the SEAs.

The mgragentd daemon is registered with the snmpdx agent. When snmpdx receives SNMP requests for the MIB supported by mgragentd, it passes the requests to mgragentd.

The configuration files for snmpdx are in the /etc/snmp/conf directory. The snmpdx daemon resides in the /usr/lib/snmp directory. Messages from the snmpdx daemon are written in the /var/adm/messages file.

To run snmpdx, enter the following command:

```
/usr/lib/snmp/snmpdx
```

The snmpdx daemon starts the mibiisa daemon.

**Note:** The snmpdx daemon and the mibiisa daemon must be running for the Tivoli NetView product to work correctly.

When you start your machine, the snmpdx and mibiisa daemons start automatically.
Chapter 2. Defining and Managing a Security Policy

You can use Tivoli NetView security services to define a security policy for your network. This chapter, which is intended for security administrators or whoever is responsible for managing Tivoli NetView security, will help you understand Tivoli NetView security services so you can define a security policy that best suits your needs. This chapter also describes what you need to do to define a security policy. You will need to perform these tasks before turning security on. You might also need to perform some of these tasks after turning security on to add new Tivoli NetView users or applications. In addition, this chapter describes the tasks to be performed in order to manage your security policy, such as how to distribute the security configuration to other servers in your network and how to view audit data.

The following topics are described:

- “Understanding Tivoli NetView Security Services”
- “Defining a Security Policy” on page 26
- “Managing Tivoli NetView User Profiles” on page 29
- “Defining the Global Security Settings” on page 34
- “Managing Security” on page 37
- “Converting ARFs to SRFs” on page 42
- “Verifying Security Permission For Shell Scripts” on page 42

Understanding Tivoli NetView Security Services

The key to effective security is understanding how the security features work and then enforcing the features. Tivoli NetView security services provides the following controls:

- Network authentication and identification
- Protected network communication
- Password protection
- Continuous, auditable network management
- Network access control of Tivoli NetView resources
- Customized Tivoli NetView graphical interface
- Audit management
- Consistent security controls
- Pager service for event correlation

Network Authentication and Identification

Security services authenticates each user, allowing access to the Tivoli NetView product when the user logs in using a valid Tivoli NetView user ID, group ID, and password. Additional login controls include restricting user access to specific days and times and to specific client and server machines. You can create a profile for each user that contains this information.

See “Managing Tivoli NetView User Profiles” on page 29 for more detailed information.
Protected Network Communication

Communication between session partners (users, clients, servers) uses a security context to ensure message integrity and identification. When the Tivoli NetView server receives a request for authentication, the server verifies the login and stores a security credential or ticket. This ticket enables shared-key security for each Tivoli NetView process when establishing a session. Shared keys are used to validate messages between session partners by checking incoming messages to ensure they are from the correct sender.

Log In Process

Users can log into the Tivoli NetView product using one of the following methods:

- Log in using the NetView Authentication Dialog by entering the `nvauth` command on the command line. The NetView Authentication Dialog is displayed, which contains input fields for the users login ID, group ID, and password.
- Log in from the command line using the `nvauth -login nvid nvgid` command. See the man page for more detailed information about the `nvauth` command.
- Start the Tivoli NetView product. If a user starts the Tivoli NetView product without first logging in, the NetView Authentication Dialog is displayed. The Tivoli NetView product is initialized after successfully logging in.
- Use the Options —> Shift_in operation on the NetView Authentication dialog box menu bar. See "Continuous, Auditable Network Management" on page 23 for a description of the Shift_in and Shift_out operations.

Users can log out using one of the following methods:

- In the Tools pull-down menu, click User Security to display the NetView Authentication dialog box. Then click the Options —> Logout operation in the NetView Authentication dialog box menu bar.
- Enter the `nvauth -logout nvid` command on the command line.
- Exit the Tivoli NetView product. When a user exits the Tivoli NetView product, the user is automatically logged out.

Logging out stops all user-initiated Tivoli NetView processes.

Refer to the Tivoli NetView for UNIX User’s Guide for Beginners for more information about login and logout.

Log In Considerations

Keep these things in mind about the Tivoli NetView login process:

- Only one Tivoli NetView login from the same OS user ID is permitted. Subsequent Tivoli NetView users from the same OS user ID can start the Tivoli NetView product without having to log in. The first user ID is the process owner for audit purposes.
- If you have a security product installed, such as DCE, you can customize your security policy so that a Tivoli NetView password is not required. Your security product manages the login procedure, allowing users to log in using a single password.

By default, the Tivoli NetView product requires a password unless you change this setting when you define the global security settings. See "Defining the Global Security Settings" on page 34 for more detailed information.
A user issuing any server-restricted operation from a client workstation does not have to be logged in as a root user and will not be prompted for the root password. The user must, however, have the appropriate security permissions and the same OS user ID on the server as on the client. Server-restricted operations are those operations that involve changing configuration files that exist on the server. These operations include:

- Event configuration (Event Configuration Trap Customization: SNMP option from the Options pull-down menu or the xnmtrap command)
- SNMP configuration (SNMP Configuration option from the Options pull-down menu or the xnmnmpconf command)
- MIB data collection (Data Collection and Thresholds: SNMP option from the Tools pull-down menu or the xnmcollect command)
- Loading and unloading MIBs (Tools —> Web Console MIB Loader , Options —> Load/Unload SNMP V1 MIBs, or the xnmloadmib command)
- Polling interval changes (Topology/Status Polling Intervals: IP option from the Options pull-down menu or the nmpolling command)
- Converting events to alerts (AIX only) (Event Configuration —> Trap to Alert Filter Control: SNMP option from the Options pull-down menu or the tralertdfc command)
- Rule set editor (Rule set Editor option from the Tools pull-down menu or the nvrseEdit command)
- Security administration (Security Administration option from the Administrate pull-down menu or the nvsec_admin command)

### Password Protection

The password, supplied when a user tries to log in, is encoded and compared with the encoded password stored in the security database. If the two match, the user gains access to the Tivoli NetView product. Passwords are not stored or sent over the network in human-readable format.

You set the user’s initial password, and you can change the user’s password at any time, when you create or change the user’s profile.

See “Creating and Changing a User Profile” on page 31 for more detailed information.

The user can also change the password at any time by selecting the Operations —> Change Password operation from the NetView Authentication dialog menu bar.

Refer to the Tivoli NetView for UNIX User’s Guide for Beginners for more information.

### Continuous, Auditable Network Management

You might require that the Tivoli NetView product manage your network without interruption. Operators using the same physical display can use the Shift_in and Shift_out operations to accomplish uninterrupted network monitoring. To use the Shift_in and Shift_out operations, operators must be in the same Tivoli NetView group and use the same OS user ID.

After the first operator logs in, the operator uses the Shift_out operation at the end of the shift. The Shift_out operation activates a window lock, allowing Tivoli NetView processes to continue. The session is protected because only an authorized operator can remove the window lock. The next shift operator removes
the window lock by clicking on the key displayed on the window lock screen to remove the window lock and then using the **Operations —> Shift_in** operation from the NetView Authentication Dialog box menu bar. Ownership of currently running Tivoli NetView processes changes to the new operator for audit and identification purposes without having to restart each process.

The operator can regain the screen saver before using the Shift_in operation by clicking the **Reset** button on the NetView Authentication Dialog box.

Refer to the *Tivoli NetView for UNIX User’s Guide for Beginners* for more detailed information about how to use the Shift_in and Shift_out operations.

**Network Access Control of Tivoli NetView Resources**

User permissions or rights to access Tivoli NetView resources are defined based on the Tivoli NetView group to which the user belongs. Each Tivoli NetView application provides a security registration file (SRF), which lists the application’s resources, such as menu items, commands, and tools. The SRF includes the valid permissions (read, write, and execute) for the application’s resources.

Vendor applications can also register with security services to make use of and integrate with security services.

The *Tivoli NetView for UNIX Programmer’s Guide* provides detailed information about integrating an application with the Tivoli NetView product’s security application.

Each Tivoli NetView group is associated with a security registration files list. The security registration files list is a collection of all the Tivoli NetView applications that a group of users can access and the group’s access permissions for all the elements within each application.

The Tivoli NetView product provides the following preconfigured groups:

- **Oper**  
  Intended for network operators. Users in this group can perform basic network monitoring tasks.

- **SrAdmin**  
  Intended for system administrators or those with more network experience. Users in this group can perform all network monitoring tasks, including advanced network problem determination, configuration changes, and security administration.

When you **create a Tivoli NetView user profile** you specify the groups to which the user belongs. This process also enables you to create a new group, thus, defining a different set of access permissions.

See “Managing Tivoli NetView User Profiles” on page 29 for more detailed information.

As an example how group permissions work, consider the following groups and their associated security registration files:

**Table 4. An Example of Group Permissions**

<table>
<thead>
<tr>
<th>User</th>
<th>Group</th>
<th>Security Registration Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>operator</td>
<td>Oper</td>
<td>ApplA, ApplB</td>
</tr>
</tbody>
</table>


Table 4. An Example of Group Permissions (continued)

<table>
<thead>
<tr>
<th>User</th>
<th>Group</th>
<th>Security Registration Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>SrAdmin</td>
<td>ApplA ApplB ApplC</td>
</tr>
</tbody>
</table>

The permissions defined for ApplA and ApplB are different depending on whether the user belongs to the Oper group or the SrAdmin group. The user, admin, in the SrAdmin group might have execute permission while the user, operator, in the Oper group might have read permission. A user in the Oper group does not have any access to ApplC; only a user in the SrAdmin group has permissions for ApplC.

**Customized Tivoli NetView Graphical Interface**

When a user logs into the Tivoli NetView product, the graphical interface presents only those functions and tools to which that user’s group has access. For example, if a group has no permissions set for a specific menu operation, the menu operation is not displayed for that group of users. If a group has read permission set for a specific menu operation, the menu operation is grayed out for that group of users.

A user can belong to more than one group, allowing you to define access control based on multiple user roles and levels of experience.

**Audit Management**

Security services collects audit data for the following security-related activities:

- Configuration changes. These include:
  - Event configuration.
  - Changes to polling intervals.
  - SNMP configuration.
  - Configuration changes defined by vendor applications.

  Refer to the *Tivoli NetView for UNIX Programmer’s Guide* for more information.

- Function access (element access)
- Login/logoff (includes Shift_in and Shift_out)

The audit data is stored in a log file, which you can review. This ability to review the audit data enables you to investigate any unusual activity that might indicate an attempted security breach. It also enables you to review normal usage patterns on your network and reliably trace security-related activities to a user, date, and time.

See “Reviewing Audit Data” on page 40 for information about how to view the audit data.

If you do not want to collect any audit data or if you do not want to collect data for all three categories, you can change this setting when you define the global security settings.

See “Defining the Global Security Settings” on page 34 for more detailed information.

**Consistent Security Controls**

You can configure your security policy on one server and distribute the same security configuration to other servers in your network. Distribution of a central security configuration provides consistent security controls and reduces
security-related administration tasks. You identify your manager station as the
distribution server when you define the global security settings.

See “Distributing the Security Configuration” on page 38 and “Defining the Global
Security Settings” on page 34 for more detailed information.

Pager Service For Event Correlation Rulesets
You can define pager information when you create a Tivoli NetView user profile
that is used to automatically issue a call to a pager. You do not need to turn
security on to make use of the pager service.

When you create an event correlation ruleset that includes a paging action, you
specify the Tivoli NetView user ID of the person to be paged. The Tivoli NetView
product looks for the pager information in that user’s profile. If the Tivoli NetView
user profile does not exist when you create the ruleset, a dialog box is displayed in
which you can enter the user ID and pertinent pager information, and the user
profile is created.

See “Creating and Editing a Ruleset” on page 136 and “Managing Tivoli NetView
User Profiles” on page 29 for more detailed information.

Defining a Security Policy

The following list describes the process you should follow to define your security
policy:
1. Create a user profile for each Tivoli NetView user. You might also want to
create new Tivoli NetView groups to create different sets of access permissions.
2. If you created new Tivoli NetView groups, define the access permissions for the
new groups.
3. Define the global security settings.
4. Test your security configuration.
5. Turn security on.

Define your security policy using the Security Administration dialog box as shown
in Figure 3 on page 27.
Accessing the Security Administration Dialog Box

By default, you must be a root user to access the Security Administration dialog box. When security is turned on, you must also be an authenticated Tivoli NetView user and log in with a valid Tivoli NetView user and group ID. Tivoli NetView permission is required to execute the Security Administration option from the Administer pull-down menu (nvsec_admin command). The SrAdmin group is preconfigured with permission for security administration.

You can change operating system file permissions so that non-root users can execute security administration. To do so, follow these steps as a root user on the Tivoli NetView server that you want to use for security administration:

Note: Umask must be equal to 022 only for a root or non-root user that sets up security using nvsec_admin.

1. Create a new operating system user group or use an existing operating system user group. The remaining steps use a group named secadmin.

2. Set group permissions for the nvsec_admin command as follows:
   \[
   \text{chmod g+x /usr/0V/bin/nvsec\_admin}
   \]

3. Set security operating system file permissions as follows:
   \[
   \text{chmod -R g+rw /usr/0V/security}
   \]

   \[
   \text{chmod -R g+rw /usr/0V/security/C}
   \]
chmod g+rx /usr/OV/security/conf
chmod g+rw /usr/OV/security/conf/sec.conf
chmod -R +rwx /usr/OV/security/C

4. To execute security administration as a non-root user from a client machine, define the same OS user ID on the server machine.

5. Add the security administration group, for example, secadmin, to the groups set for root.

To access the Security Administration dialog, enter the /usr/OV/bin/nvsec_admin command at the command line. When security is turned on, you can also access the Security Administration dialog box by clicking Security Administration from the Administer pull-down menu.

If you want the Security Administration option to be selectable when security is turned off, edit the /usr/OV/registration/C/nvauth file and remove the Security ; line from the "nvsec_admin" Action clause.

**Description of the Security Administration Dialog**

The Security Administration dialog box contains three sections:

**Users**
Lists all users known to the Tivoli NetView product and the groups to which they belong. Two sample user IDs are provided: operator and admin. The passwords are the same as the user IDs: operator and admin, respectively.

**Note:** You should change the passwords for the sample user IDs before you turn security on.

You can use the buttons to add, change, delete, and view user profiles. You can also query logged on users.

**Groups**
Lists all groups known to the Tivoli NetView product and their associated security registration files. Initially, the Oper and SrAdmin groups are listed. You can use the buttons to add, change, copy, delete, and view group permissions.

**Security Registration Files**
Lists all applications registered with security services. Select an application in the list and then click the View button to view the application’s security registration file.

**Note:** You can view only one security registration file at a time. If you have selected more than one security registration file, the View button is not available.


The menu options available when you click Options in the Security Administration menu bar enable you to define global security settings and perform other administrative tasks, such as querying all logged in users, sending messages to logged in users, forcing off a logged in user, viewing audit reports, and distributing the security configuration to other servers in your network.
Managing Tivoli NetView User Profiles

Security services authenticates each user who attempts to log into the Tivoli NetView product and permits access based on the information in the user’s profile. The user’s profile includes login information, such as the Tivoli NetView user ID, the Tivoli NetView groups to which the user belongs, and the user’s password.

The groups to which the user belongs define the user’s access permissions. When you create a Tivoli NetView user profile, you can add the user to an existing group. To create a unique set of access permissions for a user, you can create a new group. You can update a user’s profile any time, for example, to add the user to another group or include additional login controls, such as permissible login times.

Use the User dialog box as shown in Figure 4 to perform the following types of tasks:

- “Creating and Changing a User Profile” on page 31
- “Viewing a User Profile” on page 31
- “Deleting a User Profile” on page 31
- “Adding and Changing a Group” on page 31
- “Copying a Group” on page 33
- “Deleting Group Permissions” on page 34
- “Viewing a Group’s Permissions” on page 34
- “Deleting a Group” on page 34

![Figure 4. User Dialog Box](image)
Description of the User Dialog Box

The User dialog box contains three sections. The top section contains general user profile information and contains the following fields and buttons:

**NV User ID**
Mandatory field that specifies the user’s Tivoli NetView login ID.

**NV Group IDs**
Mandatory field that specifies the group or groups to which the user belongs. Type or click on the arrow button next to this field to specify one or more group names. Separate multiple group names with a comma.

**Client List**
Optional field that specifies the client host names from which the user can log in. The asterisk (*) in this field indicates permission from all clients. Specify one or more host names to restrict login from the specified clients. Separate multiple host names with a comma.

**Manager List**
Optional field that specifies the management stations to which this user can log in. The asterisk (*) in this field indicates permission to log into any server. Specify one or more host names to restrict login to the specified servers. Separate multiple host names with a comma.

**Last Profile Update**
Informational field that indicates when the user profile was updated.

**Set Password**
Optional setting to set the user’s password. You can change a user’s password any time without knowing the user’s current password. When you click the **Set Password** button, a password dialog is displayed. Enter the user’s password in both password fields and click the **Ok** button. If the password matches in both password fields, the password is set and the dialog box closes.

If you do not set the user’s password, a default password is assigned. The default password is set to the same string as the user ID. For example, if you are creating a user profile for `pamb` and do not set a password, the password is set to `pamb`.

**Set Pager Information**
Optional setting to specify the user’s pager information. You can [create an event correlation ruleset](#) that automatically issues a call to a user’s pager.

When you create a ruleset that includes a paging action, you specify the Tivoli NetView user ID of the person to be paged. The Tivoli NetView product looks for the pager information in the Tivoli NetView user’s profile.

See “Creating and Editing a Ruleset” on page 136 for more information.

When you click **Set Pager Information**, a dialog box is displayed. Enter the user’s numeric or alphanumeric pager ID in the appropriate field and use the arrow button to click the carrier name. Click **OK** to close the dialog box.

**Last Password Update**
Informational field that indicates when the password was last updated.
User IDs, Group IDs, and passwords can be from 1 to 8 characters long and can consist of the letters a through z and A through Z in addition to the following characters:

. , ; ( ) ' / = & + ; = < >

Note: Some of the allowed characters have special meaning to the operating system. Avoid using those characters that have special meaning to the operating system, especially at the beginning of an ID or password. For example, the < can be interpreted as input redirection and can cause errors when logging in from the command line.

Use the Permissible Login Time section if you want to restrict user login to a specific time and specific days. Type or use the arrow keys to set the appropriate start and end times and use the toggle buttons to select the appropriate days of the week. The default setting permits the user to log in any time on any day of the week.

Use the Notes section if you want to add comments to the user profile, such as the user’s name and phone number.

Creating and Changing a User Profile

Use this procedure if the groups to which the user belongs are already created. If you want to create a new group for a user, use the procedure described in “Adding and Changing a Group” or “Copying a Group” on page 33.

Follow these steps:
1. Access the Security Administration dialog box
2. Do one of the following:
   • If you are adding a user profile, click Add which is next to the Users section.
   • If you are changing a user profile, select the appropriate user in the Users section. Click Change.

   The User dialog box is displayed as shown in Figure 4 on page 29.
3. Make the appropriate changes to the dialog box. Use the online help if you need information about the dialog box fields. Click OK.

   The User dialog box closes, and the new user is added to the Users section of the Security Administration dialog box. The user has the permissions associated with the groups to which the user belongs.
4. Repeat steps 2 through 3 for each user profile you want to add or change.

Viewing a User Profile

You can view user profile information by selecting the appropriate user ID in the Users section of the Security Administration dialog box. Then select View.

Deleting a User Profile

To delete a user profile, select the appropriate user from the Users section of the Security Administration dialog box and then click the Delete button. Click Delete in the Delete Confirmation box to confirm this action.

Adding and Changing a Group

Adding a group enables you to customize access permissions for a particular user or set of users. You can also add a group by copying an existing group’s profile.
Minimally, you need to enable permission to run the graphical interface by setting read and execute permission for the ovw_binary and File SRFs. If you want to create a group that has minimal Tivoli NetView security permissions (permission to run the graphical interface), you might find it useful to copy the 0per group and exclude permissions to SRFs that you do not want the group to access. See “Copying a Group” on page 33 for those steps.

Changing a group enables you to customize access permissions for all the users who belong to the group. When a new application registers with security services, for example, you would change the groups that should have access to the new application, defining the permissions each group should have. If you changed a group’s permissions, users in that group will not get the change until they log off and then log back on again.

To create or change a group, follow these steps:

1. **Access the Security Administration dialog**
   - If you are changing a group, go to step 4.
   - If you are creating a group, continue to the next step.

2. Add a user to the group using one of the following methods:
   - If the user profile is already created, click the appropriate user in the Users section of the Security Administration dialog box. Then click **Change**.
   - To create a user profile, click **Add**.

   The User dialog box is displayed as shown in Figure 4 on page 29.

3. Make the appropriate changes to the dialog box fields, adding the new group name to the NV Group IDs field. Then click **Ok**. Refer to the online help for detailed information about these fields.

   The User section of the Security Administration dialog box is updated with the information, and the group name is added to the Groups section. At this point, the group does not have any access permissions.

4. Select the group name in the Groups section of the Security Administration dialog box. Then click **Add/Chg**.

   The Add/Change Group Security Registration dialog box is displayed.

5. Click the arrow button next to the Applications field and select every security registration file and click **OK**.

   **Note:** By default, if you do not select a security registration file, users have access to the elements defined in that SRF and those menu options will be displayed.

6. Click **OK** in the Add/Change Group Security Registration dialog box.

   An Add/Change Group Security Registration dialog box is displayed that contains a list of the elements of each application.
7. Click on the permissions box. 

The Element Permissions dialog box is displayed.

8. Select the appropriate permission for the element and click **OK**:
   - Click **r** to make the element unavailable. Menu options are displayed but are greyed out.
   - Click **rx** to make the element available. Menu options are displayed and are available, and command line executables are available.
   - Do not select any permissions to make the element unavailable. Menu options are not displayed.

**Note:** In general, write permission (**w**) is not used. Write permission is used only if an application’s SRF file includes **w** as a valid permission.

9. Click **Propagate** to propagate the permissions to all sub-elements (lower level menu options) or individually set the permissions for each sub-element as described in step 8.

10. Click **OK** in the Add/Change Group Security Registration dialog box. The group profile is updated with the permissions that you set.

**Copying a Group**

If you want to create a group based on an existing group’s profile, you can copy a group. Then change the group to meet your requirements.

To copy a group, follow these steps:
1. Access the Security Administration dialog box.
2. Do one of the following:
   - Add or change a user profile, entering the new Tivoli NetView group name in the Tivoli NetView Group IDs field. See “Creating and Changing a User Profile” on page 31 for those steps. Then select the group name in the Groups section of the Security Administration dialog box and click **Copy**.
   - Click **Copy** to copy a group without first adding a user to the group.
The Copy Group Profile and Permissions dialog box is displayed.

3. Use the arrow button next to the Source Group field to select the group name that you want to copy.

4. Enter the name of the new group in the Target field and click OK. The group you have just created has the same permissions as the group you copied.

5. Change the group to meet your requirements. See “Adding and Changing a Group” on page 31 for those steps.

Deleting Group Permissions
To delete group permissions, select the group name in the Groups section of the Security Administration dialog box, select the appropriate security registration file in the Security Registration Files Section, and click the Delete button. The security registration file you selected is deleted from the group’s security registration files list.

Viewing a Group’s Permissions
To view the permissions set for a group, select the group name in the Groups section of the Security Administration dialog box. Then select the appropriate security registration file in Security Registration Files section and click View.

Deleting a Group
Before you can delete a group, you must remove all users from the group. To do so, update the appropriate users’ profiles, deleting the group name from the NV Group IDs field. See “Creating and Changing a User Profile” on page 31 for those steps. When you delete the last user in the group, a delete confirmation dialog box is displayed. Click Delete to delete the group profile.

Setting User Environment Variables
You can set the environment variables, NVID and NVGID, in each user’s .profile or .kshrc file so that the user’s login ID and group ID are automatically displayed in the Authentication login panel.

For example, if Mary’s login ID is mary in the SrAdmin group, add the following lines to Mary’s .profile:

export NVID=mary
export NVGID=SrAdmin

Defining the Global Security Settings
To define the global security settings, use the Global Settings dialog box as shown in Figure 6 on page 35.
Description of the Global Settings Dialog Box

The Global Settings dialog box contains the following fields and buttons:

Security Level
Use the Change button to select one of the following settings:

ON    Turns security on for all users.
OFF   Turns security off for all users and is the default setting.
TESTMODE
Enables you to test your initial security configuration (before turning security on). Enter an OS user ID and server host name in the Test Mode ID field. Use the server host name that the operating system ID is logging into to test security. (The server host name is optional, the default is the server name you are using to change the TESTMODE global setting). You can log in as any Tivoli NetView user in any Tivoli NetView group from this OS user ID to test the configuration. Security is off for all users except for the test mode ID.

Because TESTMODE has the same effect as turning security off, test future configuration changes by logging on with a Tivoli NetView ID in the Tivoli NetView group you want to test.

Note: No other users can be logged into the Tivoli NetView product when you change the security level. If other users are logged in, a message will be displayed. See "Managing Logged In Users" on page 38 for information about how to query who is logged on and communicate with logged-in users.

Is Distribution Server
 Specifies your Tivoli NetView server as the security distribution server. When you click this button, you can distribute the security configuration to other servers in your network.

See "Distributing the Security Configuration" on page 38 for more information.

Prompt for Login
 Specifies a password prompt when logging into the Tivoli NetView product and is the default setting. If you have a third-party security product installed, you can turn this setting off so that your security product manages the login procedure, allowing users to log in using a single password.

click Audit Categories
 Specifies the audit data you want to monitor. By default, audit data is collected for the following audit categories:
• Configuration Changes
• Function Access
• Login/Logoff

See "Audit Management" on page 25 for more information about collecting audit data.

Data is logged in the log file you name in the Audit Log File Name field. You can review the audit data by using the Options —> Audit Report operation from the Security Administration dialog box menu bar.

See "Reviewing Audit Data" on page 40 for more information.

Audit Log File Name
Specifies the full path name of the audit log. The default audit log is the /usr/OV/log/sec_audit_log file.

Number of Audit Files
Specifies the number of backup audit log files to keep; the default is 3.
When the audit log file reaches the size specified in the Maximum Audit File Size field, it is moved to a backup file in the directory that is specified in the Temporary Directory field. The naming convention for the backup audit log files is as follows:

\[ \text{log\_file\_name.timestamp} \]

Where log\_file\_name is the name of the audit log file specified in the Audit Log File Name field, and timestamp indicates the date and time when the audit log file was backed up. The backup process continues until the specified number of audit files is reached. Then, the oldest backup audit log file is removed when the current audit log file is backed up.

**Maximum Audit File Size**

Specifies the maximum audit log file size in megabytes; the default is 30 MB.

**Temporary Directory**

Specifies the full path name of the directory in which to store the backup audit log files.

**Steps**

To define the global security settings, follow these steps:

1. Access the Security Administration dialog box.
2. In the Options pull-down menu on the Security Administration dialog box, click Global Settings.
   
   The Global Settings dialog is displayed.
3. Make the appropriate changes to the dialog box.
4. Click Apply to apply the changes and close the Global Settings dialog box.

**Changing the Defaults for Interaction with NetSP**

The /usr/OV/security/kkcfg file contains default settings that enable Tivoli NetView security services to interact with the IBM Network Security Program (NetSP). The following entries in the /usr/OV/security/kkcfg file enable Tivoli NetView authentication and verification:

\[ \text{comdname=/tmp/.krypto\_as\_req} \]
\[ \text{cacheroot=/tmp/NetSP\_tkt} \]

You should change these entries to reference a directory other than the /tmp directory, such as /usr/OV/tmp to ensure that security information is not inadvertently deleted. For example, some organizations run cron jobs that periodically delete all the files in the /tmp directory. Change the default settings in the /usr/OV/security/kkcfg file as follows:

\[ \text{comdname=/usr/OV/tmp/.krypto\_as\_req} \]
\[ \text{cacheroot=/usr/OV/tmp/NetSP\_tkt} \]

**Managing Security**

This section describes the administrative tasks you can perform that can help you enforce and manage your security policy. The following tasks are described:

- “Managing Logged In Users” on page 38
- “Distributing the Security Configuration” on page 38
- “Reviewing Audit Data” on page 40
Managing Logged In Users

You might find it necessary to send administrative messages, find out who is logged on, what processes are running, or force off a logged in user. For example, if a process is consuming system resources, you can determine who is running that process by querying logged in users. You can also send any administrative message to a logged in user.

Suppose you updated a group’s permissions for a new application. Users in that group will not get the change until they log off and then log back on. You might take steps similar to the following:

1. **Access the Security Administration Dialog**

2. Determine who is logged in using one of the following methods:
   - In the **Options** pull-down menu on the Security Administration dialog box, click **Query All**.
   - Select one or more users from the Users section on the Security Administration dialog. Then click **Query**.

The Logged In Users panel is displayed containing a list of logged in users.

3. Send a message to the appropriate user informing the user to log off and then log back on to access the new application.
   a. Select the appropriate user in the Logged In Users dialog box and then click **Message**.
      A message dialog is displayed.
   b. Type the appropriate message and click **Send**.

4. If the user does not respond, you can log the user off by selecting the user ID on the Logged In Users panel. Then click **Logoff**.
   The user is logged off, and a message is displayed informing you that this action is completed.

Distributing the Security Configuration

To distribute your security configuration, you must have defined your server as the distribution server on the **Global Settings** dialog box. See “Defining the Global Security Settings” on page 34 for those steps.
You can distribute your entire security configuration or selected file sets, depending on the configuration changes you have made. Follow these steps:

1. Access the Security Administration Dialog box.
2. In the Options pull-down menu on the Security Administration dialog box, click Distribution.
   The Security Distribution dialog box is displayed.
3. The section labeled Target NetView Server lists the managers that have been discovered. Select from the list or type the target server host names.
4. Select one or more file sets.
   The Users file set contains user profiles, the groups file set contains group profiles and group permissions, and the Security Registration file set contains the SRFs for each application registering with security services. After initially configuring your security policy, distribute all the file sets. Thereafter, when you make configuration changes, you can distribute only the file sets to which you made changes.
5. Click Send.
   The status window displays the transmission status for each target server, and the dialog box closes.

Security configuration files on the target servers are automatically backed up when you distribute new configuration files should you need to restore the prior configuration.

Each backup directory contains a time stamp (represented by \textit{ddmmyytime}) indicating the day, month, year, and time when the directories were backed up. A list of the file sets and the backup directory for each on the target server is as follows:

\begin{table}
\centering
\begin{tabular}{|l|l|}
\hline
\textbf{File} & \textbf{Backup Directory} \\
\hline
Users & /usr/OV/security/$\text{LANG}$/Users.ddmmyytime \\
\hline
\end{tabular}
\end{table}
If you want to restore this directory, copy the directory to the /usr/OV/security/$LANG/Users directory.

Groups
/usr/OV/security/$LANG/Groups.
If you want to restore this directory, copy the directory to the /usr/OV/security/$LANG/Groups directory.

Security Registration
Application SRFs:
/usr/OV/security/$LANG/Domains.ddmmyytime
If you want to restore this directory, copy the directory to the /usr/OV/security/$LANG/Domains/registration directory.

Group SRFs (group permissions)
/usr/OV/security/$LANG/Domains.ddmmyytime/groupname
If you want to restore this directory, copy the directory to the /usr/OV/security/$LANG/Domains/groupname directory, where groupname represents the name of the Tivoli NetView group.

Reviewing Audit Data
To view or print the audit data you are collecting, follow these steps:
1. In the Options pull-down menu on the Security Administration dialog box, click Audit Report.
   The Security Audit window is displayed.
2. View the reports, using one of the following methods:
   • To view all audit reports:
     In the View pull-down menu, click Audit Report.
   • To view specific data:
     a. In the View pull-down menu, click By Criteria. Then, in the View pull-down menu, click Set Criteria.
     The Set View Criteria dialog box is displayed.
b. Enter the search criteria and click **Apply**.

The Open Audit File dialog box is displayed containing a list of directories that contain audit files.

When you select one or more directories, the list of files contained within those directories is displayed in the Files section.

---

**Figure 9. Security Audit: Set View Criteria Dialog Box**

<table>
<thead>
<tr>
<th>Security Audit: Set View Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Search String</strong></td>
</tr>
<tr>
<td><strong>AIX Id</strong></td>
</tr>
<tr>
<td><strong>NetView Id</strong></td>
</tr>
</tbody>
</table>

Select Audit Categories:

- [ ] Configuration Changes
- [ ] Function Access
- [ ] Login/Logoff

[Apply] [Cancel] [Help]
3. Select the files you want to view and click **Open**.

You can also open, print, or save a file using the operations available when you click **File** in the Security Audit window.

**Converting ARFs to SRFs**

To convert existing Tivoli NetView application registration files (ARF) to Tivoli NetView security registration file (SRF) syntax for registering sensitive security resources, use the `/usr/OV/bin/c_arf2srf` command. See the `c_arf2srf` man page for more information.

Refer to the *Tivoli NetView for UNIX Programmer’s Guide* for information about SRF syntax and integrating applications with security services.

**Verifying Security Permission For Shell Scripts**

If you use shell scripts to run Tivoli NetView executables, use the `/usr/OV/bin/vfy_access` command to verify security permission. If you have a shell script named `myscript` that invokes the `ovobjprint` command, add the following line to the script:

```bash
vfy_access ovobjprint
```

You can specify one or more Tivoli NetView executables with the `vfy_access` command. See the `vfy_access` man page for more information.
Chapter 3. Creating and Customizing Submaps

This chapter describes the major components of the graphical interface and explains how these components work together to help you monitor and manage your network. This chapter also describes how to create a customized submap hierarchy and how to group together objects in your network. These tasks can help you monitor and manage your network more effectively.

The following topics are described in this chapter:

- "Objects"
- "Symbols"
- "Maps" on page 48
- "Submaps" on page 57
- "Using Tivoli NetView Applications" on page 64
- "Customizing a Graphical Map" on page 67
- "Defining and Managing SmartSets" on page 83

**Objects**

An object is an internal representation of a logical or physical entity or resource that exists somewhere in a computer network. An object is made up of a set of fields that specify all the characteristics of the object. Examples of resources represented by objects include:

- A computer node
- A software process on a computer
- An IP network

Most of the objects discovered and displayed by the Tivoli NetView product are network objects. However, objects can also be created by users or by applications and integrated with the Tivoli NetView product.

For information about how users can add objects, see "Adding Symbols and Objects" on page 68. For information about how applications can add and manipulate objects, refer to the Tivoli NetView for UNIX Programmer’s Guide.

**Displaying the Object Database**

Object and attribute information is stored in the object database. You can display the contents of the Tivoli NetView object database with the `ovobjprint` command.

For more information about this command, refer to the man page.

**Symbols**

A symbol is a graphical representation of an object as it is displayed on a submap of a particular map. Symbols are presentation elements; objects are underlying database elements that describe network entities like workstations, networks, and interface cards. Several symbols can represent the same object, even when the symbols are on different submaps.

Even though symbols represent objects, symbols can have some additional characteristics beyond those of the object they represent. These characteristics, or attributes, can vary among the different symbols representing a particular object.
Defining Symbol Characteristics

The following list describes characteristics of symbols in the Tivoli NetView graphical interface:

Symbol type

The symbol type consists of the symbol class, which specifies the outer shape of the symbol, and the symbol subclass, which specifies the graphic shown within the shape.

The Tivoli NetView product provides a variety of predefined symbols. To see the symbol types provided by the Tivoli NetView product, click Help —> Legend in the Tivoli NetView main menu. If necessary, you can define new symbol types using Symbol Type Registration Files. Refer to the Tivoli NetView for UNIX Programmer’s Guide for more information.

Symbol variety

A symbol can be an icon symbol, which is a two-dimensional picture, or a connection symbol, which is drawn as a line connecting two symbols. It is important to understand the difference between a connection symbol, which is a line that often represents an interface card object, and a connector symbol, which is a diamond shaped icon symbol that can represent a device like a bridge or router.

Symbol location

A symbol can reside on either the application plane or the user plane of a submap.

Submap planes are described in “Understanding Submap Planes” on page 61.

Symbol behavior

Symbol behavior defines what happens when you double-click on the symbol.

Explodable Symbols

When you double-click an explodable symbol, a child submap is opened. This is the default behavior for symbols.

Executable Symbols

When you double-click an executable symbol, the program represented by that symbol is started.

An executable symbol is displayed as a raised button on the submap. To see an example of an executable symbol, click Help —> Legend in the Tivoli NetView main menu.

Symbol label

Each symbol has a label that describes the object represented by the symbol. The label is displayed below the symbol. Because the Tivoli NetView product does not use the symbol label to identify the symbol, the symbol label does not have to be unique. You can choose whether or not to display the label of a symbol.

See “Modifying and Displaying Symbol Labels” on page 97 for steps on displaying symbol labels.

Symbol status

Symbols can display information about the status of the object or connection that the symbol represents.

Colors are used to represent status information, which is described in the Tivoli NetView for UNIX User’s Guide for Beginners.
Indicating Symbol Status

Symbol status conveys the current state of a network entity based on a predetermined set of rules. The graphical interface indicates the status of a symbol by its color on a submap. The status displayed by the symbol stems from the state of certain attributes of the object that the symbol represents. If the symbol represents a container object, for example an Internet symbol, its status color represents the combined status of all the symbols contained within it. See “Understanding Compound Status Source” on page 46 for information about compound status.

The ipmap application determines the status of a node object based on the operational status of all of the IP interfaces installed in the node. If all of a node’s IP interfaces are down, the ipmap application reflects the node’s status as critical. From the perspective of the ipmap application, the node is nonfunctional. However, another application might consider the same node to be running, because that application monitors a different protocol with fully functional interfaces. This is an example where you might have different symbols representing different states of one object.

The following list summarizes the status source used to determine status for the various symbols:

**Object Status**

Interface card symbols and connection symbols that correspond to interfaces (not networks) derive their status by object status source.

**Symbol Status**

All node symbols (connectors, servers, and computers) on network and segment submaps derive their status by symbol status source.

**Compound Status**

All other symbols on your IP submaps derive their status by compound status source. These symbols include, but are not limited to:

- All network and segment symbols
- All location and internet symbols
- All node symbols on the internet and location submaps

You can determine or change the status of a symbol by clicking **Edit Modify/Describe Symbol** in the object context menu.

Each status source is described in the following sections.

**Note:** If a symbol has object status source or symbol status source and users have configured events as status events, the Event Display application can change the symbol’s status. In addition, other applications can change the status of a symbol.

Refer to the *Tivoli NetView for UNIX Programmer’s Guide* for more information.

By default, if a symbol has compound status source, the symbol’s status is updated by the ipmap application. To enable the Event Display application to update a symbol’s status if the status source is compound, change the value for the overrideCompoundStatus resource in the `/usr/OV/app-defaults/Nvevents` file to TRUE. When the value for the
overrideCompoundStatus resource is TRUE, the Event Display application updates all symbols for the object without having to manually change the status source for each symbol.

**Understanding Object Status Source**
Object status source determines the symbol’s status based solely on the status of an individual object in the object database. If the interface card object in the object database is down, then the status is reflected as critical. If the interface card object is up, then the status is reflected as normal. The status of the interface card symbol is based on that card object only.

**Understanding Symbol Status Source**
Symbol status source determines the symbol’s status based on some algorithm different from object status source and compound status source. The algorithm is contained within the application that manages the symbol. For example, if an IP symbol has symbol status, then the ipmap application determines with its own algorithms what the status of the symbol should be. By default, the ipmap application uses symbol status for the following symbols:

- Any node symbol (connectors, servers, and computers) on a network subnet map
  The status of the connector is based on the combined status of only the interface cards that the connector has in that network, rather than the combined status of all of the interfaces in the connector.

- Any node symbols (connectors, servers, and computers) on segment submaps
  The status of the node is based on the combined status of only the interface that the node has in that segment, rather than the combined status of all of the interfaces in the node.

Some HP hubs, however, have status propagation rules because these HP hubs contain non-IP interface cards that do not report their status. The status propagation rules for these HP hubs are:

- On network submaps, the status of the hub symbol is based on the combined status of only the IP interface cards in that network.
- On segment submaps, the status of a hub symbol is determined by compound status rather than symbol status.

**Understanding Compound Status Source**
Compound status source determines how status values are propagated from symbols in child submaps to the symbol on the parent submap that represents the child submap and is based on the combined status of all of the symbols in the child submap of that symbol. For example, the status of a segment symbol is based on the combined status of all of the node symbols in the child submap of that segment.

The Tivoli NetView product uses the following rules to determine the combined status of a group of symbols:

- Default compound status
- Propagate most critical compound status
- Propagate at threshold value compound status

**Default Compound Status:** The default compound status scheme determines compound status as follows:
<table>
<thead>
<tr>
<th>A symbol's status is</th>
<th>If the symbols in the child submap meet these criteria:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>All symbols are either Normal, Acknowledged, UserStatus1, Unmanaged, or Unknown and at least one symbol is Normal, Acknowledged, or UserStatus1.</td>
</tr>
<tr>
<td>Critical</td>
<td>At least one symbol is Critical or UserStatus2 and no symbols are Normal, Acknowledged, or UserStatus1.</td>
</tr>
</tbody>
</table>
| Marginal            | Any of the following: 
|                     | - At least one symbol is Normal, Acknowledged, or UserStatus1 and at least one symbol is Marginal, Critical, Unreachable, or UserStatus2. 
|                     | - All symbols are Marginal, Unmanaged, or Unknown and at least one symbol is Marginal. |
| Unknown             | All symbols are Unknown or Unmanaged. |
| Unreachable         | All child objects are Unreachable. |

A symbol with compound status source will never have its status changed to UserStatus1, UserStatus2, or Unmanaged as a result of compound status propagation.

**Propagate Most Critical Compound Status:** The propagate most critical status scheme causes the graphical interface to propagate the status of the most critical symbol in the child submap to the submap’s symbol in the parent submap.

**Propagate at Threshold Value Compound Status:** The propagate at threshold value (0%–100%) scheme enables you to set two threshold values, marginal and critical. The marginal threshold value determines when a symbol’s status changes from normal to marginal. The critical threshold value determines when a symbol’s status changes from marginal to critical.

- Conditions for marginal status:
  - Marginal Threshold Value < % (Marginal + Critical) < Critical Threshold Value
  - If the percentage of marginal or critical symbols in the child submap of a parent object is greater than the marginal threshold value and less than the critical threshold value, then the parent object’s symbols that have compound status source will be marginal.
  - If the percentage of marginal or critical symbols in the child submap of a parent object is less than the marginal threshold value, then the parent object’s symbols that have compound status source will be normal.

- Conditions for Critical Status:
  - Critical Threshold Value < % (Marginal + Critical)
  - If the percentage of marginal or critical symbols in the child submap of a parent object is greater than the critical threshold, then the symbols of a parent object that have compound status source will be critical. If the critical threshold value is less than the marginal threshold value, then symbols that have compound status source will change directly from the normal state to the critical state.

**Setting Compound Status:** If you have a map open with read-write authorization, you can change the default compound status scheme on the current map or when you create a new map. The status setting applies to the entire map. You can’t set compound status scheme for individual submaps.
When you create a new map, you can set the compound status by clicking one of the following Compound Status buttons in the New Map dialog box:

- Default
- Propagate most critical
- Propagate at threshold values

After the map is created, you can change the status scheme by clicking Modify/Describe —> Map in the Edit pull-down menu. Click the preferred compound status button and click OK.

Maps

A map is a collection of objects stored in a map database that describe a set of network entities. They are displayed by the graphical interface for the Tivoli NetView product. The graphical interface uses these objects to draw icon symbols and connection symbols on the map’s submaps. A submap is a view, or window, that displays information stored in the map database. You do not view a map directly; instead, you view submaps that contain symbols representing objects within the map. Submaps are described in “Submaps” on page 57.

You can create, delete, or choose a map to be displayed from existing maps. You can even create several maps and control which applications operate on these maps. While you create maps and define their scope, applications dynamically update maps to reflect the state of the management environment.

Although you can create several maps, the graphical interface can have only one map open at a time. The open map is the map currently displayed by the graphical interface. This map can be updated by applications and users.

To view multiple maps simultaneously, invoke multiple instances of the graphical interface (ovw application). If there are multiple simultaneous ovw sessions with the same map open, only one session can have read-write access to the map.

Information in the Map Database

If you have a read-write map, click Modify/Describe —> Map in the Edit pull-down menu to change information about the map. Each map has the following attributes:

Name
The name of the map, which is assigned when the map is created. Each name must be unique. You can change the map name.

Root submap
The highest-level submap of the map. The root submap cannot be deleted.

Home submap
The submap that is displayed in the initial submap window when you open the map. You can assign any submap of the map as the home submap. The root submap is the default home submap.

Layout algorithm for root submap
The layout algorithm used for the root submap. The default is row/column. Once set, the layout algorithm cannot be changed.

Compound status scheme
The compound status scheme that applies to the entire map. This scheme determines how the graphical interface propagates status from symbols in
child submaps to the symbol of the parent object. You can change the compound status scheme for a map. You cannot change it for only a submap.

See “Understanding Compound Status Source” on page 46

**Configurable applications**

Any configurable map applications that are available on your system for that map. You can enable or disable these when you create a new map.

**Comments**

Any comments or notes about the map. This entry can be used to document the map’s creation date, purpose, or other information you want to store.

**Managing Maps in a Distributed Network Environment**

In a distributed network environment (client/server), the Tivoli NetView daemons run on the server, and the graphical interface applications run on the client. The client obtains event and topology status information from the daemons running on the server. A client/server configuration enables you to distribute the CPU and memory requirements to the client. You can use several clients, enabling you to divide the management tasks among more operators at one time.

The Tivoli NetView graphical interface application maintains the map database. In other words, a client application maintains the map database. Whether the map database resides on the server or the client depends on how the client was configured. The client can be configured to NFS mount the map database from the server, or it can be configured to store the map database locally.

Refer to *Tivoli NetView for UNIX Configuration Guide* for more information.

Each type of configuration has certain advantages, and each presents different network management considerations.

**When the Map Database Is NFS Mounted**

When the map database is NFS mounted, maps created on the client are stored through NFS on the server, and all users view the same set of map information. This is consistent with previous versions of the Tivoli NetView product regarding map information. The trade-off is network performance. Retrieving map database information from the server depends on network speed, bandwidth, and the size of the discovered network. Some map operations will be slower than if the maps were stored locally.

**Improving Network Performance for an NFS-Configured Client:** To decrease the impact on network performance due to retrieving map database information, use the following methods:

- Create new maps on the server rather than on the client. When the map is fully created, it can be utilized from the client. If you create the map on the client, the client contacts the daemons on the server to determine the network topology and then, through NFS, writes the information back to the server to create the map, symbols, and submaps. Creating maps on the server minimizes the amount of network traffic required to create the map.
- Investigate using read-only maps on the client. NFS write operations will not be performed, and NFS caches read operations, which improves network performance.
Updating the Host Name For NFS Mounted Maps: Because the map names and their host name locations are stored in the object database, if you change the host name of the server, you must update the host name in the object database so you can access the maps.

To update the host name of the server when all maps are on the same machine (NFS mounted), use the following command:

```
mapadmin -u newhostname
```

Replace `newhostname` with the new name of the server.

See the mapadmin man page for more information.

When the Map Database Is Stored Locally

When the map database is stored locally (on the client), better performance is obtained in retrieving map database information. However, because each client can have its own set of maps, more map administration is required. Consider the following:

- Changes you make to maps on the server are not reflected in maps on the client. If you want to make a change to all the maps, such as deleting a router, you must delete the router from each of the client maps.
- Restarting topology discovery on the server does not affect maps that might exist on the client. If the map on the client is out of date because the server’s discovery has been restarted, the map is invalid and cannot be opened. You should delete invalid maps on the client.

See “Deleting a Map that Is Not Valid from the Client” for more information.

Map names must be unique across the server’s domain, because objects get stored in the object database based on a map’s name. If a user tries to create a map with a map name that already exists on another client or on the server, the user is prompted to choose a different name.

In addition, clients and servers need to be configured with similar host name resolution procedures. Clients need to be able to resolve host name servers and other clients. If a domain name server (DNS) is used for a client or server, then DNS should be used for host name configuration for all clients and servers. Otherwise, a client can create a map and have its DNS-configured name recorded as the map owner, but a non-DNS client cannot resolve the host name.

Deleting a Map that Is Not Valid from the Client: If a map that is stored locally on the client is out of date with the map on the server, the map is invalid and cannot be opened. To delete the map that is not valid, use the following command:

```
mapadmin -r mapname
```

Replace `mapname` with the name of map that you want to delete.

You can use the `mapadmin -l` command to list all maps on the client. See the mapadmin man page for more information.

Updating the Host Name For Local Maps: Because the map names and their host name locations are stored in the object database, if you change the host name of the server or the client, you must update the host name in the object database to access the maps.
To update the host name when maps are stored on multiple machines (clients), you can use the `ovwls` to list all maps that exist and where they are located. Then use the `mapadmin -u mapname:newhostname` command to update the host name.

Assume that the `ovwls` command shows the following maps and locations:

<table>
<thead>
<tr>
<th>Map Name</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>nm014.networking.tivoli.com</td>
</tr>
<tr>
<td>default_cs</td>
<td>cs.networking.tivoli.com</td>
</tr>
<tr>
<td>mymap</td>
<td>cs.networking.tivoli.com</td>
</tr>
</tbody>
</table>

If you changed the host name for `cs.networking.tivoli.com` to `mgr1.networking.tivoli.com`, use the following commands to update the host names for the maps named `default_cs` and `mymap`:

```
mapadmin -u default_cs:mgr1.networking.tivoli.com
mapadmin -u mymap:mgr1.networking.tivoli.com
```

Removing Maps from the Client: Before removing a server connection from a client or deinstalling Tivoli NetView from the client, click Delete Map in the File pull-down menu or use the `ovw -rmmap mapname` command to remove all maps that are stored on the client. If you do not remove the maps, information is left in the object database on the server regarding the number of maps that exist and where those maps reside. This can provide an incorrect picture of the maps that exist.

Refer to Tivoli NetView for UNIX Configuration Guide for information about removing client access and deinstalling clients.

Managing Maps From the Command Line

Use the `nvmaputil` utility to work with maps from the command line to dynamically modify a Tivoli NetView map without recycling the netviewd daemon and all of its connected Web console submap explorers. You can also use this utility to determine which machines are in-scope for the scopes that are defined for Web server account access.

The `nvmaputil` utility solves the following type of problem: a misconfigured node from customer A could appear in customer B’s network. In other words, a misconfigured, multi-homed node could report one or more IP addresses that actually belong in another customer’s network. When this occurs, it is disruptive to stop the netviewd daemon and start the native console to correct the problem, because operators would lose any current submap explorer connections as soon as the netviewd daemon is stopped.

The `nvmaputil` utility provides the following function:

- Hide or Unhide an interface and its associated nodes.
- Manage or Unmanage an interface or a node.
- Delete all symbols attached to an object.

`nvmaputil` Command Parameter Descriptions

This section describes the `nvmaputil` command parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Display this help and exit</td>
</tr>
<tr>
<td>--mapname</td>
<td>The currently opened map to use</td>
</tr>
</tbody>
</table>
--delete
OVwDB selection names for which all associated symbols are to be deleted

--hide-symbol
Mapdb symbol IDs to hide (use the ovmapdump command to determine the symbol IDs.)

--unhide-symbol
Mapdb symbol IDs to unhide (use the ovmapdump command to determine the symbol IDs.)

--hide-interface-and-node
One or more IP addresses, where each IP address entry results in hiding an interface symbol and its associated segment-level node symbol. If this is an isConnector device, each IP address entry also hides its associated network-level node symbol.

--unhide-interface-and-node
One or more IP addresses, where each IP address entry results in unhiding both an interface symbol and its associated segment-level node symbol. If this is an isConnector device, each IP address entry also unhides its associated network-level node symbol.

--manage-node
One or more host names or IP addresses, where each entry corresponds to a node to be managed.

--unmanage-node
One or more host names or IP addresses, where each entry corresponds to a node to be unmanaged.

--manage-interface
One or more IP addresses, where each entry corresponds to an interface to be managed.

--unmanage-interface
One or more IP addresses, where each entry corresponds to an interface to be unmanaged.

--scopeinfo
The ScopeInfo.xml file to use. By default, the file actively being used by the Web server (/usr/OV/www/conf/ScopeInfo.xml) is used.

--dump-scopes
Dump scope information about all scopes and the in-scope OVwDB objects for these scopes.

--dump-scope
Dump scope information for a particular scope and the in-scope OVwDB objects for this scope.

--dump-definitions
If set to true, the dumped scope information contains scope definitions. The default value is true.

--dump-nodes
If set to true, the dumped scope information shows in-scope nodes. The default value is true.

--dump-node-interfaces
If set to true, the dumped scope information contains the set of in-scope interfaces for each in-scope node. This parameter is valid only when the --dump-nodes parameter is specified. The default is true.

--dump-interfaces
If set to true, the dumped scope information contains the in-scope interfaces. The default value is true.

--dump-router-overlaps
If set to true, the dumped scope information
shows in-scope router overlaps, where a router is found in more than one scope. The default value is true.

--dump-node-overlaps If set to true, the dumped scope information shows in-scope node overlaps, where a non-routing node is found in more than one scope. The default value is true.

--logconfig The log4j configuration file name.

nvmaputil Command Parameter Examples
This section provides examples of using the nvmaputil command parameters.

nvmaputil --hide-interface-and-node xxxxxxxxxx
Hide the interface symbol whose IP address is xxxxxxxxx and also hide the segment-level node symbol that this interface is attached to. If the node attached to this interface is a connecting device, a node symbol found on a network-level submap can also be hidden; this occurs only if all of the interfaces that belong to this network are hidden.

nvmaputil --delete xxxxxxxxxx
Dumps the scope information for scopes that are currently in use. This is based on the scopes currently defined in the /usr/OV/www/conf/ScopeInfo.xml file.

nvmaputil --unmanage-node xxxxxxxxxx
Unmanage the specified nodes.

nvmaputil --dump-scopes
Dumps the scope information for scopes that are currently in use; this is based on the scopes currently defined in the /usr/OV/www/conf/ScopeInfo.xml file.

nvmaputil --dump-scopes --scopeinfo /tmp/ScopeInfo.xml
Dumps the scope information for the scopes defined in the /tmp/ScopeInfo.xml file. Use this command if you are configuring scopes without stopping the Web server.

nvmaputil --dump-scope OurLan
Dumps the verbose scope information for a single scope, OurLan; this is based on the scopes currently defined in the /usr/OV/www/conf/ScopeInfo.xml file.

nvmaputil --dump-scopes --dump-definitions true --dump-nodes false --dump-interfaces false --dump-outer-overlaps false --dump-node-overlaps true
Dumps all of the scope definitions and node overlaps for the currently in-use scopes. Use this command to obtain a list of misconfigured IP addresses. Specify these IP address as arguments for the --hide-node-and-interface parameter.

Examples
The following nvmaputil examples involve two simple scopes. Both scopes are defined by a single network. The WestfordLAN scope is composed of network 10.141.242, and the LabLAN scope is composed of network 10.103.2. There are two multi-homed nodes that appear in both networks and, therefore, both scopes. These two multi-homed nodes are ptasillo.ma.ibm.tivoli.com (containing interfaces 10.141.242.70 and 10.107.2.100) and wopr.ma.ibm.tivoli.com (containing interfaces 10.141.242.71 and 10.107.2.12).

The following example dumps the scope information for the WestfordLAN:
/usr/OV/bin/nvmaputil --dump-scope WestfordLAN

Scope : WestfordLAN
Scope Definition
Resolved In Scope Networks
Network[1] : 10.141.242 (517) (Network Address: 10.141.242.0)
Resolved In Scope Nodes
Resolved In Scope Interfaces
Resolved In Scope Router Overlaps
Resolved In Scope Node Overlaps
Overlapped Scope : LabLAN
OverlappedInterface[1] : 10.107.2.100 (564)
OverlappedInterface[1] : 10.107.2.12 (566)

The following example dumps the scope information for the LabLAN:

/usr/OV/bin/nvmaputil --dump-scope LabLAN

Scope : LabLAN
Scope Definition
Resolved In Scope Networks
Resolved In Scope Nodes
Hostname[1] : 10.107.2.140 (1047)
Interface[1] : 10.107.2.139 (1064)
Interface[1] : 10.107.2.155 (1054)
Interface[1] : 10.107.2.3 (1040)
Interface[1] : 10.107.2.1 (1038)
Interface[2] : 10.107.2.5 (1050)
Interface[1] : 10.107.2.100 (564)
Interface[1] : 10.107.2.12 (566)
Resolved In Scope Interfaces
Interface[1] : 10.107.2.1 (1038)
Interface[2] : 10.107.2.3 (1040)
Interface[3] : 10.107.2.5 (1050)
Interface[8] : 10.107.2.155 (1054)
Resolved In Scope Router Overlaps
Resolved In Scope Node Overlaps
Overlapped Scope : WestfordLAN

The following example dumps all scope definitions and node overlaps for the
currently in-use scopes.

```
/usr/OV/bin/nvmaputil --dump-scopes --dump-definitions true --dump-nodes
false --dump-interfaces false --dump-router-overlaps false --dump-node-overlaps
true
```

Scope : LabLAN
Scope Definition
Resolved In Scope Networks
Resolved In Scope Node Overlaps
Overlapped Scope : WestfordLAN
Scope : WestfordLAN
Scope Definition
Resolved In Scope Networks
Network[1] : 10.141.242 (517) (Network Address: 10.141.242.0)
Resolved In Scope Node Overlaps
Overlapped Scope : LabLAN
OverlappedInterface[1] : 10.107.2.100 (564)
OverlappedInterface[1] : 10.107.2.12 (566)

Customizing Maps

You can customize maps to meet the needs of individual users. Customizing a map
enables you to:

- Allocate responsibility for managing your network among several people. For
elementary, network administrators expert in managing routers and gateways can
open a map that is configured to help manage those devices.
- Use network management applications that perform a specific type of task, for
element, data traffic or performance monitoring.
- Create a map that is a projection of an administrator’s responsibility or sphere of
influence.

You can customize the display of object information for maps you create. Several
maps can display information about the same object because maps get their
information from the same source, the ovwdb object database.

Reasons for Creating Several Network Maps

You can create several maps of your network. The following are reasons for
creating more than one map of a network.

- Sphere of influence
  You might want to have different maps for different areas of responsibility
within your network. In large or complex networks, it can become impractical
for one map to display all information about systems in your network. You can
create maps that focus on a specific set of system or nodal capabilities. For example, one map might be concerned with software maintenance, and another map with bridge management.

- **Management region**
  You might want to manage a specific portion of your network or partition information about your network. Maps can have specific constraints and characteristics, run different applications, cover different geographic areas, and set compound status differently.

- **Security**
  You might want to provide various levels of access to information about your network for security purposes. Using Tivoli NetView permissions, you can create a map of your network that cannot be edited, and another map that is identical to the first map except that it can be edited.

- **Troubleshooting**
  You might want to save a map so you can restore the map later, if necessary. It helps to keep a copy of a map before making changes that might have uncertain results.

- **Customized maps**
  You might want to combine the sphere of influence and management region to create a customized map of your network. Users can share the same map, or you can create a map for each user that combines specific management regions and specific spheres of influence. Perhaps a user wants several maps of the same network, each focusing on a different domain or organization. You can customize a map so that it is a projection of your responsibility for systems on your network. You can customize each map to display certain aspects of your network and avoid sifting through data you do not need. You can also use different background graphics.

- **System-specific maps**
  You might want to have separate maps for different kinds of systems on your network. For example, you might want one map for your IP systems, one map for OSI-based systems, one map for Apollo Domain systems, one map for NFS, and one map for diskless systems.

### Assigning Map Access Levels

Assigning map access levels enables you to limit or deny access on a per-user, per-map basis. Each user of your system will have one of the following types of access to each map:

- **No access**
  The user cannot open this map.

- **Read-only access**
  The user can see status changes, perform locate operations on objects, and update topological changes using the File —> Refresh Map operation. However, the user cannot add, delete, or modify symbols, objects, or submaps.

- **Read-write access**
  The user can add objects, add connections, create submaps, and change object attribute values.

### Changing Map Permissions

Only one user can have a specific map open with read-write access at one time. If you have a map open with read-write access and another user displays the same map, the other user’s map is open with read-only access. This is the case even if
the other user has read-write permission to the map. Several users cannot have simultaneous read-write access to a specific map. However, if each user creates a copy of this map, they can have read-write access to the copy.

There are several commands that you can use to assign users read-write or read-only access to a map. To change map permissions, use the `ovwperms` command or one of the following routines:

- `ovwlsl` Lists current permissions for specified maps.
- `ovwchown` Changes the owner of one or more maps.
- `ovwchgrp` Changes the group ID of one or more maps.
- `ovwchmod` Changes map permissions by mode. Do not use the file permission commands from your operating system.

For more information about these commands, refer to the Tivoli NetView for UNIX Administrator's Reference or the man pages. For steps on changing the map permissions using the Tivoli desktop, see “Setting Map Permissions” on page 98.

**Map Snapshots**

A snapshot is a static image of a particular map that preserves the status of all symbols and contains all submaps that existed in the map at the time the snapshot was taken. Although you must take snapshots from a read-write map, the snapshots are read-only and cannot be updated by applications. In the File pull-down menu, click **Map Snapshot —> Create** to take a snapshot of a map.

Use snapshots to document your network or to keep a record of your network’s current status. It is useful to take snapshots before making major configuration changes. In the File pull-down menu, click **Map Snapshot —> Open** to open a previously created snapshot of a map.

When you open a snapshot, the home submap at the time the snapshot was taken is displayed. Only one snapshot can be open at a time; however, you can display the open map and a snapshot in submap windows at the same time. The name of the snapshot is displayed in the status line.

Although you can display an open map and a snapshot of the map at the same time, you cannot perform the same operations on them. Operations that highlight a symbol on the open map do not highlight the same symbol on the snapshot. Highlighting an object applies only to the map or snapshot in which it is highlighted.

**Submaps**

A submap is a collection of related symbols that are displayed in a single window. Each submap displays a different perspective of the information in the map. Typically, submaps are organized in a hierarchy that enables you to see your network from a distance or to choose a more detailed view. You can customize the organization of submaps in a map to suit your purposes.

The most common method used to navigate through submaps is double-clicking the mouse on explodable symbols. Double-clicking an explodable symbol causes a
Working with Submaps

Submaps enable you to:

- Create a selective view into part of the management domain of a network.
- Choose a collection of symbols to display in a single submap window.

You can create, delete, and modify the characteristics of submaps in the open map. You might want to create a submap to display a detailed view of systems on your network. You can create increasingly detailed submaps of your network map. If you have a submap that becomes too congested, you can create a new submap and partition the information.

Using the Root Submap

The graphical interface creates a root submap that provides a standard, top-level submap on which you can display the symbols that represent different protocol views of the map. For example, on the root submap you will probably have an IP Internet symbol that represents all of the IP entities in your management domain. You can also have symbols that represent other types of topology entities. The root submap enables you to place more than one protocol symbol within one map.

Network and systems management applications can use the root submap to build hierarchies of submaps. The root submap serves as an anchor on which applications can place symbols that represent protocols. You can select one of these symbols and display the highest level of a submap hierarchy.

Using the Home Submap

Each map has a submap designated as the home submap. The home submap is the first submap that is displayed when the map is opened. You can assign any submap in the map as the home submap for all users of that map. By default, when a map is created, the root submap is designated as the home submap. If you delete the home submap, the root submap becomes the home submap until another home submap is assigned.

See “Assigning a Home Submap” on page 80 for more information.

Submap Characteristics

Table 6 describes submap characteristics.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the submap, as assigned when the submap was created. Each name must be unique within the scope of the map.</td>
</tr>
<tr>
<td>Parent Object</td>
<td>The object considered a parent object of a submap. The symbols representing the object explode into the submap. Because several symbols from different submaps can represent the same object, you can navigate to a child submap from several submaps. A submap might not have a parent object.</td>
</tr>
</tbody>
</table>
Table 6. Submap Characteristics (continued)

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Submap</td>
<td>The submap chosen as the parent of the current submap. During submap creation, a user or application can choose a submap to be the parent submap.</td>
</tr>
<tr>
<td>Layout</td>
<td>The layout algorithm used for the submap. Once the algorithm is set, it cannot be changed for an existing submap.</td>
</tr>
<tr>
<td>Presentation</td>
<td>The presentation format used by the Tivoli NetView product to display the submap. The presentation can be either scaling or zooming.</td>
</tr>
<tr>
<td>Background Graphics</td>
<td>The graphic displayed on the background of the submap plane to customize the appearance of the submap.</td>
</tr>
<tr>
<td>Comments</td>
<td>The comments, notes, or keywords about the submap.</td>
</tr>
</tbody>
</table>

Creating Child Submaps and Independent Submaps

In the Tivoli NetView product, there are child submaps and independent submaps. The method you use to create a new submap determines whether the submap is a child submap or an independent submap (also known as an orphan submap).

A child submap represents a detailed view of its parent object. To create a child submap, double-click an explodable symbol whose parent object has no child submap.

For more information about creating child submaps, see “Creating a Child Submap” on page 78.

An independent submap has no parent object or parent submap. To create an independent submap, click Create Submap in the Edit pull-down menu. To display an independent submap, click the submap from the Submaps in Map dialog box and click Open Submap, or click the independent submap from the Navigation Tree window.

Understanding Submap Presentation

Submap presentation enables you to choose how each submap presents symbols and background graphics. You can choose between scaling and zooming for each specific submap.

**Scaling**

Scaling enables you to display an overall view of the submap. All displayed symbols and the background graphic scale to the size of the submap window. Scaling is the default setting when a submap is created.

**Zooming**

Zooming enables you to display a close-up view of the submap. Scroll bars are displayed when only a portion of the submap fits into the viewing area. If you are working with a read-write submap and you use the scroll bars to change the visible portion of the submap, the layout is saved.

You can use the zoom feature using the following methods:

- Set a zoom factor to determine the extent to which you zoom into the view of the submap. The default zoom factor is one.
• Use quick zoom to draw a boundary box around the area of the submap to be magnified.

**To set a zoom factor:**
1. In the Edit pull-down menu click **Modify/Describe**.
2. Click **Submap**.
3. Click the Zooming button and use the slider bar to select a zoom factor.
4. Click **OK**.

**To use quick zoom:**
1. Position the mouse cursor in the upper left corner of the area to be magnified.
2. Press and hold down the Shift key and mouse button 1. Then drag the mouse to draw a box around the area to be magnified.
3. Release mouse button 1, then the Shift key. If you release the Shift key first, the objects will be selected instead of magnified. Use the **Locate —> Selected Objects List —> Deselect All** menu operation to deselect them and try again.

To return to the original presentation, press and hold down the Shift key and click mouse button 1 anywhere on the map.

Displaying a submap with background graphics and zoom presentation uses a large amount of memory. This is because the entire virtual display has to be buffered, enabling the user to navigate using the slide bars. The checkZoomAllocation resource in the /usr/OV/app-defaults/OVw file controls whether a warning dialog box is displayed every time a user zooms in on a submap with background graphics. Zoom ratios greater than the value set for checkZoomAllocation cause a warning dialog box to be displayed. Set the resource to 10 if you do not want a warning dialog box to be displayed. Set the resource to zero if you want a warning dialog box to be displayed for any zoom ratio.

See "Changing the Graphical Interface Defaults" on page 99 for more information.

**Understanding Submap Layouts**

The way the Tivoli NetView product arranges symbols on a submap is called the submap layout. The method for arranging symbols on the submap is called the layout algorithm. Symbols can be automatically placed on a submap as determined by the layout algorithm or they can be manually placed by the user.

**Using Layout Algorithms**

Each submap has an assigned layout algorithm that determines how symbols are arranged on the submap. The layout algorithms are based on common network topologies. Table 7 lists the available layout algorithms.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Arrangement on the Submap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row/Column</td>
<td>Symbols are arranged in rows and columns.</td>
</tr>
<tr>
<td>Point to Point</td>
<td>Symbols are arranged as an arbitrarily inter-connected set of nodes and connections.</td>
</tr>
<tr>
<td>Bus</td>
<td>Symbols are arranged along a backbone representing the linear array of nodes on a segment.</td>
</tr>
<tr>
<td>Star</td>
<td>Symbols are arranged in a star consisting of a circle and a center symbol. You can set the star center using the symbol pop-up menu.</td>
</tr>
</tbody>
</table>
Table 7. Network Topology Layout Algorithms (continued)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Arrangement on the Submap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ring</td>
<td>Symbols are arranged in a circle.</td>
</tr>
<tr>
<td>Tree</td>
<td>Symbols are arranged in a hierarchical tree structure.</td>
</tr>
<tr>
<td>No layout</td>
<td>Symbols are arranged by the user or are left in the New Object Holding Area.</td>
</tr>
</tbody>
</table>

You can set the layout algorithm for a submap only when the submap is created. The layout algorithm cannot be changed for that submap after it has been created. If an application creates a submap, it can specify a layout algorithm. If no layout algorithm is specified when the map is created, a default layout algorithm is selected. The default layout algorithm for a submap is based on the symbol type of the parent object.

**Note:** You cannot change the layout algorithm for any submap of the system default map. The default layout algorithm for the root submap is row/column.

**Using Automatic Layout**

Automatic layout either enables or disables enforcement of the layout algorithm of a submap. You can enable or disable the automatic layout algorithm for a selected submap. After enabling or disabling automatic layout for all submaps in the open map, you can change the automatic layout setting to on or off for all submaps. See “Setting Automatic Layout” on page 97 for steps on setting automatic layout.

**Using the New Object Holding Area**

A New Object Holding Area is displayed in the lower portion of each submap window if the submap has no layout algorithm or if automatic layout is disabled for that submap and new objects have been discovered. Symbols in the New Object Holding Area are shown without their connections. To drag symbols from the new object holding area and place them in the associated submap, hold the Ctrl key and using mouse button 2, select the symbol, drag it to the submap, and release the mouse button.

**Understanding Submap Planes**

Submaps contain the following three layers, or planes:

- Background plane
- Application plane
- User plane

The **background plane** provides the background against which symbols are viewed. You can add background graphics in the background plane to provide a context for looking at symbols in your submap presentation. Symbols on the application plane and the user plane are displayed on top of the graphic.

For more information about adding background graphics, see “Adding or Removing a Background Graphic” on page 95.

Symbols on the **application plane** represent objects that are managed by at least one network or system management application. If one or more applications manage an object, one or more symbols of that object are displayed on the application plane. The symbols of that object are presented flat against the submap background. If no applications are managing the object, all symbols representing that object appear on the user plane.
Symbols on the user plane represent objects that are created by users and not managed by any applications. The Tivoli NetView product distinguishes symbols on the user plane by providing a shadow for symbols to make them appear raised above the submap background.

**Metaconnection Submap**

A metaconnection symbol represents more than one connection between two symbols or a symbol and a backbone on a submap. For example, suppose a gateway has more than one interface card in the same network. On the IP Internet submap, the connection between the gateway and the network is a metaconnection. This is because the connection symbol between the gateway and the network represents two connections (the two interface cards). When you double-click a metaconnection symbol, the metaconnection submap opens.

The metaconnection submap displays the status of each connection between the two symbols. The graphical interface creates a metaconnection submap when a user or an application adds a second connection between two symbols or a symbol and a backbone.

You can add an unlimited number of connections between two symbols in a regular submap. Each of these connections is automatically added to the metaconnection submap with the two symbols or the symbol and the backbone as endpoints.

It is possible to create a regular child submap for a connection object, if only one connection exists. If that connection becomes multiple, a metaconnection submap is created. You can no longer access the regular child submap by double-clicking the connection. However, you can click the regular child submap from the Submaps in the Map dialog box. You cannot see individual connections between the two symbols on the submap in which the metaconnection symbol is displayed.
Characteristics of a Metaconnection Submap
A metaconnection submap has the following characteristics:

- Displays all the connections represented by the metaconnection symbol.
- Has a row/column layout unless the connections are between a symbol and a backbone.
- Displays the two end points of the connection in the metaconnection submap for each connection in the submap.

Behavior of a Metaconnection Submap
The behavior of the metaconnection submap is similar to a regular submap in some ways:

- You can create a child submap from a metaconnection submap by double-clicking on any of the objects in the metaconnection submap. Therefore, the metaconnection submap can be a parent of other regular submaps.
- You can Click objects in the metaconnection submap.
- You can add unconnected objects to a metaconnection submap.

The behavior of the metaconnection submap differs from a regular submap in some ways:

- You cannot add connections to a metaconnection submap.
- You cannot delete the last object from the submap (whether it represents a connection or not) without deleting the metaconnection and metaconnection submap.
You cannot see propagated status for connected symbols. Compound Status works differently in metaconnection submaps. The metaconnection symbol displays the compound status of the multiple connections in the metaconnection submap. Any unconnected objects in the metaconnection submap also contribute to compound status. However, the connected icon symbol in a metaconnection submap do not propagate their status. Their status is maintained by the symbols in the parent submap above the metaconnection submap.

---

**Using Tivoli NetView Applications**

An application is a program that interacts with users through the graphical interface. Applications enable you to perform the following actions:

- Process user requests
- Create or delete objects, symbols, and submaps
- Change the contents of maps
- Provide special display functions

You can write and integrate applications with the Tivoli NetView product.

Refer to the *Tivoli NetView for UNIX Programmer's Guide* for more information about writing applications.

**ipmap Application**

The ipmap application is the primary application used by the Tivoli NetView product. When the graphical interface is started, the ipmap application is automatically started. The ipmap application ensures that the ovw application (the graphical interface) and the ovtopmd daemon (the IP topology database daemon) behave consistently.

For example, when an object is deleted using the graphical interface, the ovw application tells ipmap which symbols and objects were removed. The ipmap application then tells ovtopmd to make the appropriate changes to the topology database.

When netmon discovers a new node, ovtopmd adds the node to the IP topology database and informs ipmap that a new node has been discovered. The ipmap application uses what it knows about IP devices to tell ovw which icon and connection symbols it needs to create. The graphical interface then displays the correct symbols and modifies the map database accordingly.

For more information about the netmon and ovtopmd daemons, refer to "Background Processes” on page 7 or the man pages.

**Map Synchronization**

When a map that uses the ipmap application is opened, the application starts its synchronization phase. While the ipmap application is synchronizing, the graphical interface displays the [Synchronizing] message on the status line of all displayed submaps of the open map. During this phase, the ipmap application requests information about changes to the IP topology database since the map was last open. This information is continually updated by the ovtopmd daemon. If there are any new objects in the IP topology database, the ipmap application tells the ovw application which icons and connection symbols to add to the map.

The map will enter the synchronization phase for a short time during operations that change the contents of the map, for example, changing interface labels, adding new objects, or cutting and pasting large numbers of symbols.
Limitations: While the imap application is synchronizing, the following limitations exist:

- You cannot delete symbols, objects, or submaps.
- You cannot add objects.
- You cannot cut and paste symbols.
- The imap application will not appear in the Configurable Applications list on dialog boxes (Object Description).
- Manage Objects and Unmanage Objects operations will not take effect.
- Acknowledge and unacknowledge operations will not take effect.

When the synchronization phase completes, the imap application resumes full operation and the [Synchronizing] message is no longer displayed.

Using imap Application Submaps

The imap application and the graphical interface have rules that control the placement of symbols on submaps. Only certain symbols can exist on each submap type.

Table 8 describes the submap hierarchy used by the imap application and the symbols are supported on each submap.

<table>
<thead>
<tr>
<th>Submap</th>
<th>Symbols Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root submap</td>
<td>• IP Internet</td>
</tr>
<tr>
<td>Location or Internet</td>
<td>• Location&lt;br&gt;• Internet&lt;br&gt;• IP network&lt;br&gt;• Connector (for example, a gateway)&lt;br&gt;• Connection</td>
</tr>
<tr>
<td>Network</td>
<td>• Segment&lt;br&gt;• Connector (for example, a gateway)&lt;br&gt;• Connection</td>
</tr>
<tr>
<td>Segment</td>
<td>• Node (for example, computers and connectors)&lt;br&gt;• Backbone&lt;br&gt;• Connection</td>
</tr>
<tr>
<td>Node</td>
<td>• Interface card</td>
</tr>
</tbody>
</table>

Moving through the Submaps

Double-clicking the IP Internet symbol takes you into the IP Internet submap, which is a special internet submap. The internet submap might contain a few network symbols. Double-clicking a network symbol takes you into a network submap. On the network submap, you will probably have segment symbols. Double-clicking a segment symbol takes you into a segment submap. The segment submap will have nodes connected to a backbone symbol. Double-clicking the node symbol displays a node submap. The node submap displays all of the interfaces contained in the node.

Each time you select a symbol, you get a submap that is more specific than the previous submap. You can view your entire network, a part of the network, or a single node. Each time you view the parent of a submap a more expansive view of your network is displayed.
**Root Submap:** The root submap is the highest level submap. It is at the top of the Navigation Tree and has no parent. The root submap contains the IP Internet symbol created by the ipmap application. The IP Internet symbol is the only symbol on the root submap that the ipmap application manages. If you have other applications that use the graphical interface to draw symbols, you can have other symbols on the root submap, but ipmap does not know of their existence.

**Internet and Location Submaps:** Internet and Location submaps show logical groupings of IP networks and subnetworks connected by gateways. Location submaps and Internet submaps are considered to be equivalent by ipmap. This means that any symbol type that can be placed on a location submap can also be placed on an Internet submap.

The IP Internet submap is special because it is created by the ipmap application. It is also the highest level Internet submap; it is the only Internet submap that does not have a location or internet object as its parent. The root submap is its parent. The ipmap application places discovered IP addressable gateways and networks on this submap.

Internet and location objects are often referred to as container objects. That is because Internet and Location submaps are the only submaps that can contain their own symbol type. For example, a location submap can contain other location symbols, but a node submap cannot contain other node symbols. It can only contain interface symbols. This function can be used to organize and simplify a map. This is called partitioning. See [“Partitioning Submaps” on page 74](#) for information about partitioning.

**Network Submaps:** The network submap represents the physical topology of a network at the level of network segments. The ipmap application can discover and display IP-addressable segments, gateways (routers), repeaters, multiports (hubs), bridges, and the connections between them on the network submap.

**Note:** If you are using IP subnetting, network implies subnet.

**Segment Submaps:** A segment submap represents the physical topology of a segment of your network at the level of nodes and connectors. It displays the computers and connectors that comprise a segment on your network.

In IP networks, a segment is a group of data communication objects that are interconnected through a common transmission medium. Nodes belonging to the same segment typically use a common physical medium to communicate with each other (for example, Ethernet, token ring, telephone lines, or satellite links). The following segment topologies can be drawn:

- **Bus** Represents nodes attached to a single linear cable that transmits data (for example, Ethernet or IEEE 802.3)

- **Token ring** Represents nodes attached to an SNMP, IP-addressable token ring central wiring MAU through twisted pair wiring, which conforms to the IEEE 802.5 standard

- **Star** Represents all nodes attached to an SNMP central multiport repeater (a hub)

- **FDDI ring** Represents nodes attached to an SNMP, IP-addressable Fiber Optic Data Distribution Interface (FDDI)
Node Submaps: The node submap displays symbols that represent the components of a node in a row/column layout. The graphical interface draws interface symbols on the node submap. When an interface object is added to a map, the ipmap application tells the graphical interface on which node submap to draw the interface card symbol. The ipmap application also tells the graphical interface on which higher level submaps to draw connection symbols.

Using the xxmap Application

The xxmap application enables you to see submaps showing information about open-topology objects. The xxmap application presents information gathered by the gtmd daemon. The gtmd daemon stores and correlates topology information received in the form of SNMP traps or through API calls based on the Tivoli NetView topology MIB. For more information about the gtmd daemon, refer to the Tivoli NetView for UNIX Programmer’s Guide or the man page.

The xxmap application supports user changes to the map. Using the graphical interface, you can add and delete symbols. However, added symbols are not verified or stored in the gtm database. These symbols exist only in the user plane. When you delete a symbol from the map, the underlying object is not deleted from the gtm topology database. Objects can be deleted or added to the gtm database with an SNMP trap command or an API call. This symbol can also reappear during a subsequent map synchronization phase. If you do not want to see the symbol on a map, click Hide Objects in the Edit pull-down menu to hide it. The object is still managed even though it is hidden. To view a list of objects that are hidden, click Hidden Objects List in the Edit pull-down menu.

Displaying Multiple Protocols

If you are using protocols other than IP, the xxmap application enables you to see a list of the managed protocols running on a selected object. This option is available only for objects that are interfaces or nodes.

Suppose you select an object that is a computer running the CMIP protocol. Select Protocols in the View pull-down menu to display a list that contains the following information about the CMIP protocol interface on the selected node:

- The name or address as defined in the Tivoli NetView General Topology MIB
- The status of the protocol
- Submaps that contain the selected object’s CMIP interface

Customizing a Graphical Map

There are many ways in which maps can be customized to make them easier to use and understand. Objects can be added to and deleted from your map database, and symbols can be moved from one submap to another. You can also group objects together into a SmartSet based on a selection rule that you define, such as a SmartSet of all routers.

Submap characteristics such as submap name, parent submap, layout, and background graphics can be changed. Also, objects can be selectively managed and acknowledged to match your management region responsibilities. The following sections describe how to use these customization functions.

Customizing Symbol Placement

You might want to customize the IP Internet topology to reflect your network layout: geographically, hierarchically, or by some other criteria that is important to
your company. For example, you could add location symbols to your IP Internet submap that correspond to your offices in New York, Chicago, and San Francisco and group all networks under these location symbols. You can do this manually or you can create a custom configuration file to automatically customize your network layout. This type of customization is called submap partitioning.

Submap partitioning involves many different actions, including adding objects and moving symbols. First, you should know the rules for manually adding, deleting, moving, and copying symbols and objects. To automate this level of customization in your maps using the location file, refer to “Automating Internet Submap Partitioning” on page 77.

Note: A symbol can be an icon symbol, which is a two-dimensional picture, or a connection symbol, which is drawn as a line connecting two icon symbols. It is important to understand the difference between a connection symbol, which is a line that often represents an interface card object, and a connector symbol, which is a diamond shaped icon symbol that represents a device such as a bridge or router. Throughout this section the term symbol refers to an icon symbol unless stated differently.

Adding Symbols and Objects
The Tivoli NetView product enables you to add symbols to submaps. The following are some of the reasons why you might want to do add symbols:
• A network entity might not have been automatically discovered by the netmon network discovery daemon.
• You want to add something to your map that is not a part of your network, such as a location or segment symbol.

If you know the object you want to add and on which submap you want to place the symbol, you can drag the symbol from a palette of symbols to the correct submap and then enter some information about the object. See “Steps for Adding Objects and Symbols” on page 70 for the steps on adding objects.

Note: Connection symbols are added differently from icon symbols. See “Adding Connection Symbols and Objects” on page 70 for information about adding connection symbols.

User Plane and Application Plane: A symbol that has a shadow is in the user plane, and a symbol that has no shadow is in the application plane. A symbol in the application plane is known and controlled by ipmap or some other application, whereas a symbol in the user plane is known and controlled by only the user that added or modified the symbol. Click User Plane —> For This Submap —> Shadow Off in the View pull-down menu to turn the shadow off an object in the user plane.

Symbols Managed by the IP Application: Certain applications only manage certain kinds of symbols. The ipmap application can manage the following types of symbols, organized by symbol class:

<table>
<thead>
<tr>
<th>Symbol Class</th>
<th>Managed Symbols</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cards</td>
<td>IP interface card</td>
</tr>
<tr>
<td>Computer</td>
<td>All</td>
</tr>
<tr>
<td>Connector</td>
<td>All</td>
</tr>
</tbody>
</table>
If you want the ipmap application to manage an IP entity, then use one of the symbols listed above. If you do not use one of the symbols listed above, then the symbols will not have status and connectivity information updated by the ipmap application.

**Symbols That Can Be Added:** The ipmap application enforces rules that control the placement of symbols on submaps. The ipmap application only manages symbols if they are placed on the correct submap. This ensures that submaps can always be traversed. It also ensures that connectivity and other relationships between objects can be conveyed properly.

Table 10 describes symbols that can be added to the submaps and still be managed by the ipmap application. It is organized by submap type. The table also lists what information is needed about an object before it can be added.

**Table 9. Symbols Managed by ipmap (continued)**

<table>
<thead>
<tr>
<th>Symbol Class</th>
<th>Managed Symbols</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>All</td>
</tr>
<tr>
<td>Network</td>
<td>Internet, IP Network, Bus, Token Ring, Star, FDDI, Serial, and Frame Relay segment</td>
</tr>
</tbody>
</table>

**Table 10. Symbols You Can Add That ipmap Can Manage**

<table>
<thead>
<tr>
<th>Submap</th>
<th>Symbols That Can Be Added</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root</td>
<td>None.</td>
</tr>
<tr>
<td></td>
<td>The IP Internet symbol is the only symbol controlled by ipmap on this submap. Because this symbol is created for you, there is no reason to add any others.</td>
</tr>
<tr>
<td>Location or Internet submap</td>
<td>• IP network symbol (network class). You must provide unique network name, IP address, and subnet mask. • Gateway symbol (connector class). You must provide unique host name, IP address, and subnet mask. • All location symbols. You must provide unique selection name. • Internet symbol (network class). You must provide unique selection name.</td>
</tr>
<tr>
<td>Network submap</td>
<td>• All connector symbols. Provide a unique host name, IP address, and subnet mask. • Bus, star, token ring, FDDI, serial, frame relay symbols (network class). Provide a unique selection name.</td>
</tr>
<tr>
<td>Segment Submap</td>
<td>• All connector symbols. Provide a unique host name, IP address, and subnet mask. • All computer symbols. Provide a unique host name, IP address, and subnet mask.</td>
</tr>
<tr>
<td>Node Submap</td>
<td>IP interface card symbol. Provide an IP address and subnet mask.</td>
</tr>
</tbody>
</table>
Steps for Adding Objects and Symbols

Now that you know what and where symbols can be added, follow these steps to add an object and symbol. You must have the map open with read-write access to add objects.

1. In the Edit pull-down menu, click Add —> Object.
2. Click a class from the Symbol Classes on the Add Object Palette.
3. Using mouse button 2, drag the desired symbol subclass icon to the appropriate submap and release the mouse button.
4. When the Add Object dialog box is displayed, complete the following fields:
   - Label
   - Display Label
   - Behavior
   - Object Attributes
   - Selection Name
   - Comments
5. If you want your added object to be updated, controlled, and managed by an application, then you have to:
   a. Click the application name in the Object Attributes list on the Add Object dialog box. If the symbol represents an IP entity, select the IP Map list item.
   b. Click Set Object Attributes.
   c. Fill in all the necessary fields.
6. Click OK in the Add Object dialog box to add the symbol to the submap and close the dialog box.
7. Click OK in the Add Object Palette to close the palette.

If this is done correctly, after you drag the symbol to the submap, you will see that the symbol is displayed flush against the background plane, and the symbol will be drawn without a shadow. The symbol is in the application plane. This indicates that the application you specified is correctly managing the symbol you added.

If you do not select an application and fill in the correct fields, the symbol is displayed raised above the background plane, and a shadow is displayed under the symbol. This means that the symbol is in the user plane and receives no updates from any existing applications.

The application that manages IP entities is called the ipmap application. When adding IP objects to your map, selecting the IP Map entry in the Object Attributes list and selecting the Set Object Attributes button ensures that your IP symbols will correctly reflect status and connectivity information.

Adding Connection Symbols and Objects

To draw a connection between two symbols, click Edit —> Add —> Connection. When you connect two IP symbols together, you are specifying that a new IP interface card exists that connects the symbols. For example, when you connect a gateway to a backbone symbol, you are adding an interface card to the gateway. The connection symbol is a graphical representation of the interface object in the object database.

Symbols That Can Be Connected: Table 11 on page 71 describes which IP symbols can be connected together with connection symbols. It is organized by submap type.
Table 11. Symbols That Can Be Connected

<table>
<thead>
<tr>
<th>Submap</th>
<th>Symbols That Can Be Connected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root</td>
<td>None</td>
</tr>
<tr>
<td>Location or Internet submap</td>
<td>Gateway symbol to IP network symbol</td>
</tr>
<tr>
<td>Network Submap</td>
<td>Gateway symbol to a bus, star, token ring, FDDI, serial, or frame relay segment</td>
</tr>
<tr>
<td>Segment Submap</td>
<td>All connector symbols to backbone symbol</td>
</tr>
<tr>
<td>Node Submap</td>
<td>None</td>
</tr>
</tbody>
</table>

**Steps for Connecting Symbols**

Now that you know what symbols can be connected, follow these steps to add a connection. You must have the map open with read-write access.

1. In the Edit pull-down menu, click **Add —> Connection**.
2. In the Add Connection palette, click one of the connection symbol types.
3. Click a source symbol on the submap that you want to connect.
4. Click a destination symbol. The connection symbol is displayed between the two selected symbols on the map.
5. When the Add Object dialog box is displayed, complete the following fields:
   - Label
   - Display Label
   - Behavior
   - Object Attributes
   - Selection Name
   - Comments
6. If you want your connection symbol to be updated, controlled, and managed by an application, then you must:
   a. Click the application name in the Object Attributes list on the Add Object dialog box. If the connection symbol represents an IP entity, click **IP Map**.
   b. Click **Set Object Attributes** button.
   c. Complete the remaining fields.

   If this procedure is not used, the connection symbol is not updated.

7. Click **OK**.
8. Click **OK** in the Add Object dialog box to set the connection in the map and close the dialog box.
9. Click **OK** in the Add Connection Palette to close the palette.

The application that manages IP entities is the ipmap application. When connecting IP symbols on your map, click the IP Map entry in the Object Attributes list and click the **Set Object Attributes** button. This ensures that your connection symbols correctly reflect the status of the underlying interface card.

**Deleting Objects and Symbols**

To delete an object from the map, click **Delete Object —> From All Submaps** in the Edit pull-down menu. To delete an object completely from the object database, go into every map that contains that object and delete the object from all submaps. Click **Delete Object —> From All Submaps** to be sure all symbols for an object are deleted from your map.
If the object is still part of your network, and if the netmon daemon is running, a deleted object can be discovered and displayed. There are three ways to prevent this:

- Turn off new node discovery by clicking **Topology/Status Polling Interfaces: IP** in the **Options** pull-down menu.
- Prevent discovery by starting the netmon daemon with a seed file that contains the IP address or host name of the node preceded by an exclamation mark (!). The ! operator tells netmon not to discover this node. See the netmon man page for more information.
- Let the object be discovered and then click **Hide Objects** in the **Edit** pull-down menu to hide the object. The object is still managed even though it is hidden. To view a list of objects that are hidden, click **Hidden Objects List** in the **Edit** pull-down menu.

You must have a map open with read-write authorization to delete an object or symbol. You cannot retrieve deleted objects or symbols.

Do not use the **Edit —> Delete Object** menu to delete all nodes in the internet view. The complete deletion of the database while the Tivoli NetView graphical interface is running can cause unpredictable results.

To delete all objects in the map, shut down all Tivoli NetView graphical interfaces and use the Server Setup application (serversetup). Click one of the following options:

- **Maintain —> Clear databases —> Clear object/topology/map databases, save customizations**
- **Maintain —> Clear databases —> Clear object/topology/map databases, remove customizations**
- **Control —> Restart automatic map generation**

### Moving Objects and Symbols

The **Edit —> Cut** and the **Edit —> Paste** functions are used to move symbols from one submap to another. The graphical interface lets you move any symbol you want to any submap you want. Although the graphical interface enables you to move any symbol, some symbols might be managed by particular applications that place limits on where symbols can be placed.

For example, the ipmap application only manages IP network symbols that are located on location or internet submaps. If an IP network symbol that is managed by ipmap is cut from a location submap and pasted onto a node level submap, then the ipmap application will stop managing that symbol. For this reason, it is a good idea to be familiar with the rules that each application places on the location of certain symbols.

As a general rule, if a symbol is in the application plane (the symbol has no shadow and is controlled by an application) and you move (cut and paste) the symbol, you want the symbol to end up in the application plane of the destination submap.

If you want the ipmap application to continue to control an IP symbol, then move (cut and paste) IP symbols to submaps that ipmap supports.

**Symbols You Can Move:** Table 12 on page 73 describes which IP symbols can be moved and still be managed by the ipmap application. They are listed by submap type.
Table 12. Symbols That Can Be Moved and Managed by ipmap

<table>
<thead>
<tr>
<th>Submap</th>
<th>Symbols That Can Be Moved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root submap</td>
<td>None.</td>
</tr>
<tr>
<td></td>
<td>The IP Internet symbol is the only symbol controlled by ipmap on this submap. This symbol must remain on the root submap.</td>
</tr>
</tbody>
</table>
| Location or Internet submap | • IP network symbols (network class)  
• Gateway symbols (connector class)  
IP Network and gateway symbols can be moved from one Location or Internet submap to another Location or Internet submap. They should not be moved to any other type of submap. |
| Network submap              | None.                                                                                     |
|                             | No symbols should be moved to or from a network submap. That is because the symbols on a Network submap have IP addresses and they would be inconsistent with the IP subnets of other networks. |
| Segment submap              | • All connector symbols  
• All computer symbols  
All connector and computer symbols can be moved between segments in the same network. Node symbols should not be moved between segments in different networks. |
| Node submap                 | None.                                                                                     |
|                             | No symbols should be moved to or from a Node submap.                                      |

When using the cut function to move symbols from one submap to another, always click the option **Cut —> From This Submap**. Cutting from all submaps will cause multiple copies of your symbol to be moved, and that could cause links between symbols to be drawn incorrectly.

The cut and paste options enable you to move symbols from one submap to another. They should not be used to copy symbols.

**Cut Buffer:** The cut buffer holds symbols you have cut or copied until you do one of the following:  
• Store another symbol into the cut buffer.  
• Open another map.  
• Exit the Tivoli NetView product.

**Steps For Moving an Object:** Now that you know what can be moved and where it can be moved to, follow these steps to move an object. You must have the map open with read-write access to move objects.

1. Open the submap that has the symbol representing the objects you want to move.
2. Click the symbols.
3. In that submap, click **Cut —> From This Submap** in the **Edit** pull-down menu.
4. Open the destination submap.
5. Click **Paste** in the **Edit** pull-down menu.

When symbols are pasted into a submap, they are first put in the user plane. Any applications that control or manage the symbols determine if they have been moved to a legal location. Depending on the machine speed and the number of symbols moved, this could take from 1 to 15 seconds. Once necessary applications determine that the move is supported, the symbols are moved into the application plane and the shadow disappears.

**Copying Symbols**

Like cutting and pasting, copying is a function that is provided by the graphical interface. Applications that use the graphical interface to control and display symbols provide various levels of support for the copy function.

The ipmap application does not support the copy function. When manipulating IP topology symbols, it is recommended that the **Edit —> Copy** function not be used. If you want to move symbols between submaps, it is best to use the **Edit —> Cut** function. IP symbols that are copied will be placed in the user plane rather than the application plane. These symbols will not show a connection to any other symbol. In general, the copy menu option on the graphical interface exists for use with other applications, and not with the ipmap application.

**Steps for Copying Symbols:** There are two parts to this procedure:

- Copying a symbol
- Pasting a symbol

1. **Copy the symbols.**
   - a. Select the symbol that you want to copy.
   - b. In the **Edit** pull-down menu, click **Copy**.
   - c. In the **Copy** pull-down menu click one of the following:
     - From This Submap
       Performs the operation on this submap.
     - From All Submaps
       Performs the operation on all submaps.

     When you click **Copy**, the symbol remains on the submap.

2. **Paste the symbols.**
   Click the submap on which you want to paste the symbol. In the **Edit** pull-down menu, click **Paste** to paste the symbol onto the submap.

   **Note:** You can paste the symbol onto any submap of the open map. Pasting the copied symbol onto the same submap from which you copied it creates multiple symbols of the object for the same submap.

   All symbols are pasted in the **New Object Holding Area** for each submap in which automatic layout is disabled.

**Partitioning Submaps**

You can partition a submap to subdivide submaps into smaller, more manageable units. For example, you can add location symbols that correspond to your offices in New York, Chicago, and San Francisco to your IP Internet Submap. You can
then move the network and gateway symbols from the IP internet submap to the New York, Chicago, and San Francisco submaps, making network entities much easier to find.

There are two kinds of submaps that you can partition:
- Internet and location
- Segment

The rules for each are slightly different.

This section describes how to partition submaps manually. You can automate this process for the Internet submap using the location configuration file. Refer to "Automating Internet Submap Partitioning" on page 77 for more information on automating this customization.

Partitioning Internet/Location Submaps
Partition Internet or Location submaps when you want to take symbols from an Internet or Location submap and place them in other Internet or Location submaps. To partition an Internet or Location submap manually, follow these steps:

1. Add either a location object or an internet object to either a location submap or Internet submap. If you are working with IP symbols, do not forget to fill out the correct IP Map information in the Add Object Dialog Box. See "Steps for Adding Objects and Symbols" on page 70 for steps on adding objects.

2. After the location or internet object has been properly added, click the network or gateway symbols that you want to move into the new location.

3. In the Edit pull-down menu, click Cut —> From This Submap. When you have selected the symbols and cut them, they should disappear in the original submap.

4. Open the submap of the internet or location symbol you added in the first step.

5. In the Edit pull-down menu, click Paste.

Example of a Partitioned Internet Submap
Figure 12 on page 76 shows an Internet submap in which four container objects were added. All symbols of objects discovered by the ipmap application have been cut in the Internet submap and placed in lower-level partitioned Internet submaps. If you clicked on the United States symbol, you would open the submap shown in Figure 13 on page 77.
Figure 13 on page 77 shows the United States submap. Six container objects have been created in this submap. From this submap, you can double-click any symbol and open a child submap to display other container objects or symbols that represent IP networks and gateways.
Partitioning Segment Submaps
Partition segment submaps when you want to move node symbols from one segment submap to another segment submap. To partition a segment submap, follow these steps:

1. Open the network submap that contains the segment that you want to partition.
2. Add a segment object to the Network submap. If you are working with IP symbols, do not forget to fill out the IP Map dialog box when adding the segment. See “Steps for Adding Objects and Symbols” on page 70 for steps on adding objects.
3. When the new segment has been added, open the segment submap that contains the symbols you want to move.
4. Select the symbols.
5. In the Edit pull-down menu, click Cut → From This Submap. When the cut is complete, the symbols should disappear.
6. Open the new segment submap and in the Edit pull-down menu, click Paste.

Automating Internet Submap Partitioning
You can also customize the IP Internet topology to reflect your network layout automatically using the location file (/usr/OV/conf/location.conf). The location file contains the following:

- A list of networks (or ranges of networks)
- A list of routers
- The locations under which networks and routers should be displayed
Locations can be nested to create a hierarchy. All other symbols, such as segments, are placed under the network in which they belong. Locations cannot be used to divide a subnet.

Using this file, instead of manually customizing the submap, automates the customization process and will preserve the customizations when the map is regenerated.

See the Tivoli NetView Release Notes, Version 7.1 or the /usr/OV/conf/location.conf file for more information about the format of the location.conf file and how to use it.

Creating a Child Submap

Create a child submap if you need a more detailed view of an object. You can use the default settings or modified settings. Open the submap from any explodable symbol of the parent object. The submap is part of the map’s hierarchy of submaps.

If the symbol you select already has a child submap, then click Add —> Object in the Edit pull-down menu to add an explodable object. This function adds a parent object from which you can add a child submap. Use the Location symbol type to represent the object if you are adding network views.

Using Default Settings

If you have a submap that consists of three groups of objects, organize your network by creating three child submaps, then click Cut and Paste in the Edit pull-down menu to cut and paste each group of objects into a different child submap. Because you are only reorganizing your network, you do not need to change any of the default settings for the child submap.

Steps: To create a child submap with default settings, follow these steps. You must have the map open with read-write access.

1. Double-click an explodable symbol that does not contain a child submap. A Tivoli NetView Question box displays a message that the object does not have a child submap.

2. To create a child submap with default settings, click OK. The new submap is created and displayed in a separate submap window. Because this is a new submap, the submap does not contain symbols or objects.

   When you double-click an explodable symbol of the parent object, the submap is opened. When you click Show Parent on the background context menu of the child submap, the parent submap is reopened in the separate submap window.

Default Settings: The default settings for the child submap are:

- The name of the submap is the same as the selection name of the object from which the submap was created.
- The presentation is scaling.
- The submap contains no background graphic.
- The submap contains no comments.

Using Modified Settings

If you have a submap that consists of three groups of objects that represent three parts of your network and you want to change the layout of one of the groups, you might consider creating three child submaps and changing the layout.
algorithm for one of the child submaps. After you create the child submaps and change the settings as necessary, click Cut and Paste in the Edit pull-down menu to cut and paste each group of objects into the appropriate child submap.

**Steps**

To create a child submap with modified settings, follow these steps. You must have the map open with read-write access.

1. Double-click on an explodable symbol that does not contain a child submap. A Tivoli NetView Windows Question box is displayed, telling you that the object does not have a child submap.

2. To create a child submap with modified settings, click Modify. The New Submap dialog box is displayed. The name of the new submap is displayed in the Name field. The name of the parent object is displayed in the Parent Object field. The button to the right of the Parent Submap displays the name of the parent submap.

3. You can modify the following settings:
   - Name
   - Parent submap
   - Layout
   - Presentation
   - Background graphics
   - Comments

   For more information about modifying these settings, see “Modifying Submap Settings” on page 80.

4. After you complete the dialog box, click OK. The new submap is created and displayed. Because this is a new submap, the submap does not contain symbols or objects.

**Creating an Independent Submap**

You can create an independent submap (also known as an orphan submap) that does not have associated parent objects. This type of submap is independent of the existing submap hierarchy. For example, you might create an independent submap that contains all your routers. This enables you to monitor router status from one submap.

To open an independent submap, click Open Submap in the View pull-down menu. You can open a submap created without a parent object through the Navigation Tree window only if you have already opened that submap during the current Tivoli NetView session. Use the horizontal scroll bar on the Navigation Tree window to see the independent submap symbols.

**Steps**

To create an independent submap, follow these steps. You must have the map open with read-write access.

1. In the Edit pull-down menu, click Create Submap.

2. A Tivoli NetView Question box is displayed. Click one of the following buttons:
   - Click OK to create a submap with default settings. For more information about default settings, see “Creating a Child Submap” on page 78
   - Click Modify to create a submap with modified settings.
     For more information about modified settings, see “Using Modified Settings” on page 78
   - Click Cancel to undo the submap creation.
Note: Because this submap is an independent submap, you cannot assign it a parent submap.

Changing a Parent Submap
You can change the parent submap of the open submap. The parent submap is the submap whose icon is displayed last in the submap stack.

Steps
To change the parent of a submap, follow these steps. You must have the map open with read-write access.
1. Decide which open submap you want to modify. Then make it the current window.
2. In the Edit pull-down menu, click Modify/Describe —> Submap.
3. The Submap Description dialog box is displayed. Click the Parent Submap option button to display a list of possible parents for this submap.
4. Click on a parent in the list. If no list is displayed, the current parent submap is the only choice at this time.
5. After you select a parent submap in the list, click OK to apply the changes to the submap and close the dialog box.

Modifying Submap Settings
You can modify certain characteristics or information about a submap such as its presentation, the background graphic, or its parent.

Steps
To modify a submap, follow these steps. You must have the map open with read-write access.
1. Decide which open submap you want to modify. Then, make it the current window.
2. In the Edit pull-down menu, click Modify/Describe —> Submap.
3. The Submap Description dialog box is displayed. You can modify any of the following characteristics:
   - Name
   - Parent object
   - Parent submap
   - Presentation
   - Background graphics
   - Comments

Note: You cannot modify the submap layout. The layout algorithm can only be set during submap creation.
4. After you modify the characteristics on the dialog box, click OK to apply the changes to the submap and close the dialog box.

Assigning a Home Submap
To assign the submap to be the home submap that will be displayed when the map is opened, follow these steps. You must have the map open with read-write access.
1. In the Options pull-down menu, click Set Home Submap.
2. Click the submap you want to be the Home Submap on the Submaps in the Map dialog box. You can use the Find Submap field to locate a specific submap by matching a string or substring with a submap entry.
3. Click Set as Home.
4. Click Close in the Submaps in the Map dialog box to apply the change and close the dialog box.

The selected submap is the submap displayed when this map is opened.

**Modifying an Object Description**

To modify an object description, follow these steps. You must have the map open with read-write access.
1. Click one or more objects in the submap.
2. In the Edit pull-down menu, click Modify/Describe —> Object.
3. An Object Description dialog box is displayed for each selected object. You can modify the following attributes in the Object Description dialog box:
   - Selection name
   - Object attributes list
   - Comments
4. Click OK on the Object description dialog box to apply the changes and close the dialog box.

**Managing and Unmanaging Objects**

An object can be managed or unmanaged. A managed object is being monitored for topology, status, and configuration changes. The symbol for the managed object reports the status changes by changing to the color that represents the status. If an object is not being managed, the symbol for the object does not report the status because it is not known. However, the symbol is still visible. Managing objects uses network resources. If you have objects that do not need to be managed, unmanage them.

If you want an object to continue being managed but you do not want to see it, click Hide Objects in the Edit pull-down menu. The object receives and reports status but the symbol for the object does not appear on the submaps. You can also access the Hide operation by clicking Hide Objects in the Edit pull-down menu or Edit —> Hide —> Symbol in the object context menu. In the Edit pull-down menu, click Hidden Objects List to view a list of objects that are hidden.

**Steps**

To manage or unmanage one or more objects, open the map with read-write access and follow these steps:
1. Click one or more objects to be managed or to be unmanaged.
2. In the Options pull-down menu, click Manage Objects or Unmanage Objects.

All selected objects and child submaps are managed or unmanaged, depending on which option you selected.

**Acknowledging and Unacknowledging Objects**

If you know that an object has stopped functioning, but you do not want the Tivoli NetView product to notify you continuously about this problem, use the acknowledge operation.

When you acknowledge the object, the object changes to a dark green color and remains in the acknowledged state until you select the object and unacknowledge it. Unacknowledging an object causes the Tivoli NetView product to resume normal processing.
The Tivoli NetView product offers two modes of acknowledging objects: a map-based mode and a global-based mode.

Map-based mode is the default unless you have run the tecits_upgrade script, in which case the NVMAPGLOBALACK is set to Global mode. The Global Acknowledge function provided by IPMAP is designed to work only with those objects in the map created and managed by IPMAP. Global Acknowledge for non-IP map objects created by the user or other OVw applications is not supported.

You can set an environment variable to activate the global-based behavior for individual clients. You might want the default map-based behavior for some clients and the global-based behavior for others. For instance, you might have NetView client users who do not need to see objects being acknowledged by the network operations center.

Setting Map-Based Acknowledge Status
The default map-based Acknowledge mode only affects the current map. Operators of client machines must refresh the map to be notified that objects have been acknowledged or unacknowledged. Also, acknowledging objects on read-only maps is not persistent; restarting the console clears all the acknowledged statuses. Note that the Acknowledged status can not be set in the command line.

To acknowledge or unacknowledge one or more objects, follow these steps. You must have the map open with read-write access.

1. Click one or more objects to be acknowledged or unacknowledged.
2. In the Options pull-down menu, click Acknowledge or Unacknowledge.

All selected objects and child submaps are acknowledged or unacknowledged, depending on which option you selected.

Setting Global-based Acknowledge Status
In the global-based Acknowledge mode, the Acknowledged status is an object attribute. All NetView clients are notified immediately when an object is acknowledged or unacknowledged, regardless of which map is open and even if the action is performed on a read-only map. In addition, you can set the Acknowledged status of an object in the command line, permitting the transfer of the Acknowledged status to backed-up NetView databases.

To activate the global-based mode, set the environment variable NVMAPGLOBALACK equal to 1 on the NetView machine (NVMAPGLOBALACK=1). This must be set before you invoke the NetView console.

You can acknowledge an object using one of three methods. The first, in the map, uses the graphical user interface. The other two methods, from an event or through nvdbimport, allow users greater control of the Acknowledged state outside of the graphical user interface.

- **In the map.** To acknowledge or unacknowledge map objects:
  1. Select one or more objects.
  2. Click Options—> Acknowledge to set the isAcknowledged field to TRUE for that object and generate an event to update all open maps (local or NFS) connected to the NetView server.

If a NetView console running in global-based mode starts up later, the object is displayed as Acknowledged.
• **From an event.** To acknowledge or unacknowledge an object from the command line, type:

```
   event -b openview -e event -a object ID
```

Where `event` is either `ACK_EV` (acknowledge event) or `UNACK_EV` (unacknowledge event), and `object ID` is the object ID of the object you are acknowledging in the OVw database. For more information on this command, see the `event` man page.

This event sets the status of the specified object to Acknowledged on all open maps where the status of the object is CRITICAL or MARGINAL. It also sets the Acknowledged field to TRUE in open NetView maps. If no NetView sessions are running in global-based mode, this field is not set. If NetView consoles running in global-based mode start up later, the object shows as Acknowledged.

• **Using nvdbimport.** To directly set the isAcknowledged field for chosen objects in the OVw database, use nvdbimport. See the `nvdbimport` man page for more details.

---

**Import/Export Utility for Acknowledged Status**

The sample import/export utility script, `/usr/OV/bin/acknowledgeUtil`, provides an example of setting the acknowledged state from an event. To use the script, type:

```
   acknowledgeUtil -export -map mapname > filename
   acknowledgeUtil -import < filename
```

The beginning of the script includes a complete description of the arguments.

Only Critical and Marginal objects are changed to Acknowledged. Attempts to Acknowledge normal objects during the import operation are silently ignored.

---

**Defining and Managing SmartSets**

Network administrators often discover that, as their networks grow, distributing files and customization changes to nodes in the network is a time-consuming and error-prone task. The Tivoli NetView SmartSet facility provides a mechanism for you to group objects together. This group of objects is called a *SmartSet*. You can define SmartSets using the SmartSet Editor or the `nvUtil` command. You can also use the SmartSet facility APIs to have your applications create and use SmartSets.

The collmap application provides the display of SmartSets and is automatically started when you start the graphical interface. You can use the **Administer —> Start Application —> collmap** menu option to start the collmap application without having to close and restart the graphical interface. You might find this useful if the collmap application ends abnormally after you have started the graphical interface.

Note that collmap is a viewing tool only. It cannot be used to update SmartSets in the SmartSet facility. Thus, cut and paste are not supported within SmartSet submaps.

SmartSets are aggregated under a SmartSet icon and are displayed on the Root submap. If you double-click on the SmartSet icon on the Root submap, you see a submap containing symbols for all the SmartSets that are defined. Double-clicking on one of the SmartSet symbols opens a submap containing all the objects that are currently in that SmartSet. As objects move in and out of the SmartSet, the submap
is dynamically updated. A user in read-only mode needs to click **Refresh Map** in the **File** pull-down menu to see new objects that have been dynamically updated.

Defining a SmartSet can be useful for creating a submap of devices that you want to monitor closely. For example, you can define a SmartSet of all critical routers. This enables you to view a submap of all routers that are inactive at any time.

SmartSets enable you to do policy-based management. For example, you can establish a **SmartSet policy** for all routers, establish distribution policies through the **Agent Policy Manager** for specific domains as defined by SmartSets, and establish polling policies for groups of machines. You can also **create event rulesets** that check to see if the node is a member of a specified SmartSet.

If you have the Tivoli NetView MLM product installed in your network, you can use the **Agent Policy Manager** to set thresholds and set up file monitoring for SmartSets you have defined. For example, you can define a threshold to monitor CPU utilization for all objects in a Fileserver SmartSet. The Agent Policy Manager is closely integrated with the SmartSet facility. **Management by policy** in this manner facilitates your task of system management by centralizing control. If you want to change the thresholding on a group of objects, you do not have to change each object in the SmartSet; the change is applied automatically to all objects in the SmartSet. Similarly, if an object that fits the defined SmartSet is added to or taken out of the network, no additional changes are necessary. The SmartSet facility automatically updates the SmartSet.

See Chapter 8, “Using the Agent Policy Manager (APM),” on page 219 for more information.

### Managing and Unmanaging SmartSets

The SmartSet icon and SmartSet symbols are unmanaged until you double-click the icons. Double-clicking a SmartSet symbol starts the monitoring process, and status is propagated upwards from objects within the SmartSet. Because monitoring status uses network resources, you should unmanage SmartSets that do not need to be managed.

To stop monitoring SmartSet status, select the SmartSet symbol and then click **Edit** —> **Delete** —> **Symbol** from the context menu to delete the SmartSet symbol. Because you have not deleted the SmartSet definition, the symbol is automatically recreated and is unmanaged. If you delete the SmartSet icon on the IP root map, all the SmartSets under the SmartSet icon become unmanaged. If you want to unmanage a specific SmartSet and continue to manage other SmartSets, delete the SmartSet symbol for the SmartSet that you want to unmanage.

### Types of SmartSets

To create a SmartSet, you can specify the selection name of an object, or you can define a **rule** (much like a filter rule), using the Tivoli NetView object capability definitions. If you specify a rule (such as `isRouter=True`), the SmartSet facility locates objects that fit that description. A SmartSet can also be a combination of a node list, rules, and other SmartSets you have defined.

The **Agent Policy Manager** automatically creates one SmartSet for you, and you will see it appearing on your root map with the label, MLM subnets. This special collection is made up of the MLMs in your network that the Tivoli NetView product knows about, and all the objects managed by the MLMs. If you double-click this symbol, the Tivoli NetView product displays a map showing each
of the MLMs, and double-clicking an MLM symbol displays a star configuration of the MLM and the managed objects in its subnet.

See "APM SmartSet Icons That You Get Automatically" on page 220 for more information.

Quick Refresher Course on Boolean Logic

When you define a SmartSet, you have several opportunities to use logical AND and OR statements to join different rules you have specified. These logical operators have a different meaning than and and or in everyday speech. Consider this example:

- If I said to you, “Please bring me an apple and an orange,” you would return with two pieces of fruit, an apple and an orange.
- If I tell the SmartSet facility, “Locate a mainframe computer AND a PC,” it will find nothing, because there probably is no individual device that has both of those characteristics assigned to it.

The logical operator OR also operates differently:

- If I said to you, “Please bring me an apple or an orange,” you would return with either an apple or an orange.
- If I tell the SmartSet facility, “Locate a PC OR a mainframe,” it will find all the PCs and all the mainframes in the network.

When you use the SmartSet facility, keep these simple rules in mind:

- When you want to find the union of two characteristics, use OR.
- When you want to find the intersection of two characteristics, use AND.

Pattern Matching

Pattern matching enables you to use regular expressions as the value of a SmartSet rule. A valid regular expression consists of printable characters. However, in a regular expression, the following symbols have special meaning:

- . (period)
  Matches printable and nonprintable character except <newline> (unless it is used inside brackets). For example, node in matches:
  node3interface
  nodeXinterface

- * (asterisk)
  Means zero or more occurrences of the preceding character. For example, node* could match the following strings:
  nod
  node
  nodeTwo
  nodee

The expression node.* could match node1 and node, but the expression would not match nod.

The rule sysLocation~Building* would match the following strings:
  Building 002
  Building 500
  Building 062
**Note:** The ~ character is the "like" operator.

- `^` (caret)
  If the `^` character is the first symbol, the following character is the first character in the string, as shown in the following example:
  ^any Matches the string: any of them.
  ^any Does not match the string: many of them.

- `$`
  If `$` character is the last symbol, the preceding character is the last character in the string, as shown in the following example:
  long$ Matches the string: long
  long$ Does not match the string: longer

Pattern matching on IP addresses is treated differently (not as a regular expression). If the value of a SmartSet rule looks like an IP address, the syntax for using wildcards is as follows:

- **IP address:** `<byte>.<byte>.<byte>.<byte>`
- **byte:** `int | int-int | [int-int] | *`
- **int:** `0 - 255`

The rule IP Address~146.83.[120-255].* specifies all nodes with IP addresses 146.83.120.* through 146.83.255.*, where * represents any integer in the range 0–255.

**Adding a New SmartSet**

To define a new SmartSet, use the SmartSet Editor or the `/usr/OV/bin/nvUtil` command. Refer to the man page for information about the nvUtil command.

You can use a list of IP addresses, capability rules, or a combination of both to define a SmartSet. You can use pattern matching in the values of capability rules as explained in "Pattern Matching" on page 85. You can also use the name of another SmartSet, thus building a hierarchy of SmartSet.

The following example demonstrates how you can set up a SmartSet for critical routers in your network. For this example, we will set up a SmartSet that includes all of the Cisco and IBM routers. We will use the Tivoli NetView predefined object capabilities to find objects that are classified as IBM routers or Cisco routers. We will add other routers by specifying their IP addresses. In addition, we will find those routers whose IP status is critical.

1. SmartSets are defined using the SmartSet Editor. In the Tools pull-down menu, click SmartSet Editor. The SmartSet Editor window is displayed.
2. Click the Add button to display the Add SmartSet dialog.
3. Enter the name of the SmartSet, with a brief description of what the SmartSet contains. This SmartSet is called CriticalRouters.
4. Define the first SmartSet rule. Next to Definition 1, click Modify. The Modify Definition dialog box is displayed.
5. In this dialog box, click Definition Type. You will see a list of the types of rules you can use to define a SmartSet:
6. Select **Attribute**.

   You will see a list of Object Attributes. Use the scroll bar to move down through the list until you find **vendor**.

7. Select the vendor object attribute. A list of selectable values is displayed:
8. Select the IBM value and click **OK**. The definition is added to the Add SmartSet dialog box:

![SmartSet Editor Object Attributes and Values](image)

*Figure 15. SmartSet Editor Object Attributes and Values*
9. You now have a rule to select objects that have IBM specified as the value for their vendor attributes. To complete the first rule, you need to OR this rule with a rule selecting objects that have a vendor attribute of Cisco. (In other words, we will find objects that are made by IBM as well as objects that are made by Cisco.) Between Definition 1 and Definition 2, make sure the Or radio button is selected.

10. Click Modify next to Definition 2. Repeat the steps for selecting the vendor attribute, but this time select Cisco as the value for the attribute.

11. Click OK to add the definition.

12. You now have a rule to find IBM OR Cisco objects. You need to AND this rule with a rule that will find objects that are classified as routers. Ensure the And radio button between the top two definitions (1 and 2) and the bottom two definitions (3 and 4) on the Add SmartSet dialog box is selected.
13. Select the Modify button next to Definition 3. Select the Definition Type button, and select Attribute from the menu that is displayed.

14. Scroll through the list until you find the attribute isRouter. Select this attribute.

15. This time, you will see that the attribute can be set to True or False. Select the True radio button and click OK.

16. You now have a rule to find IBM and Cisco routers. You need to AND this rule with a rule that will find routers that are down.

   Ensure the And radio button between Definition 3 and Definition 4 is selected.

17. Click Modify next to Definition type 4. Select the Definition Type option button, and select Attribute from the menu that is displayed.

18. Scroll through the list until you find the attribute IP Status. Select this attribute.

19. Select the Critical value and click OK.

20. You now have a rule to find all critical routers. Here is how the completed rule looks:
21. Click **OK** to add this definition. The SmartSet you defined appears on the SmartSet Editor main window.

### Modifying a SmartSet

Suppose you need to add one or more routers to your SmartSet of critical routers. This router is a vendor that is not listed as a selectable value for the vendor attribute, but you do know the router’s IP address. You can create another SmartSet, using the CriticalRouters SmartSet you already defined, and add the new router by specifying its selection name. Here are the steps:

1. Start the SmartSet Editor if it is not running. In the **Tools** pull-down menu, click **SmartSet Editor**. The SmartSet Editor window is displayed.
2. Click **Add** to display the Add SmartSet dialog box.
3. Enter a name and description for this new rule. We will call this example SmartSet CriticalRoutersPlus1.
4. Next to Definition 1, click **Modify**. The Modify Definition dialog box is displayed. Select Include SmartSet Rule.

5. A list of defined SmartSet rules is displayed. Select CriticalRouters and click **OK**. The SmartSet name is displayed in Definition 1.

6. Next to Definition 2, click **Modify**. The Modify Definition dialog box is displayed. This time, we will use Object List as the Definition Type. It is probably already selected; if not, select Object List.

7. At the bottom of the Modify Definition window, enter the selection name of the object to be added in the Object Selection Name field and click **Add**. The object is added to the List of Objects.

8. Click **OK**. The object is displayed in Definition 2.

9. Click **And** between Definitions 1 and 2 and the **Or** radio button between Definition Definitions 3 and 4.

10. Next to Definition 3, click **Modify**. The Modify Definition dialog box is displayed. Select Attribute as the Definition Type.

11. Select the attribute IP Status and the value **Critical**.

12. Click **OK**. The IP status is displayed in Definition 3.

13. Click **OK** to add this definition. The SmartSet you defined is displayed on the SmartSet Editor main window.

**Listing Objects in a SmartSet**

To find out which objects in the network are included in a SmartSet, select a SmartSet on the SmartSet Editor window and then click **Resolve**. A list of objects in the selected SmartSet is displayed. You can also use the `/usr/OV/bin/nvUtil` command to list objects in a specific SmartSet. Refer to the man page for more information.

**More Examples of SmartSets**

The following examples illustrate other SmartSets you might develop. Each example has a statement of the rules under which an object is to be included in the SmartSet and a picture of how the rule is to be defined in the SmartSet facility.

**Example 1**

Create a SmartSet of all printers running OS/2® TCP/IP SNMP agents that are not located in building 123 or building 456.
Example 2
Make a SmartSet of all nodes contained in subnet 195.88.31.0.
Figure 19. Nodes Contained in Subnet 195.88.31.0
Chapter 4. Customizing the Graphical Interface

Customizing the graphical interface can mean as little as adding backgrounds or it can include specifying maps for startup, authorizing file permissions, setting event filters, and customizing menu bars.

This chapter describes the tasks involved in customizing the graphical interface. You can change the graphical interface for each user so they see only what they are responsible for. For example, if you have three users, each responsible for a different geographical region, you can create or customize submaps for each user so they see only the region for which they are responsible. You can give them access to applications that pertain to only them. If you want to add a different background to each submap, you can do that also.

This chapter includes the following tasks:

- "Adding or Removing a Background Graphic"
- "Arranging Symbols" on page 96
- "Assigning Maps" on page 98
- "Setting Map Permissions" on page 98
- "Customizing the Menu Bar and Tool Palette" on page 98
- "Changing the Graphical Interface Defaults" on page 99
- "Customizing the Failing Resource Display" on page 101
- "Customizing Event Filters for Users" on page 102
- "Customizing the Tivoli NetView Grapher" on page 102

Adding or Removing a Background Graphic

You can add a graphic to the background of a submap or replace a graphic with a different graphic. Background graphics can provide contextual information, such as:

- A floor plan of your business
- A geographic map showing diverse sites
- A diagram representing some characteristics of a portion of the managed network

Supported Formats

The graphical interface supports the following file formats for background graphics:

- GIF  CompuServe Graphics Interchange Format Background graphics must be in GIF87a format.
- XBM  X11 monochrome bitmap format

Note: To support the operation of sizing a window, the ovw application zooms graphics down to a minimum size (about icon size) first and then attaches the graphics to windows. If the graphic is large and contains small text and thin lines, some information might be lost when the graphic is sized down. If you can clearly view the graphic when it is zoomed down to icon size, the Tivoli NetView product should clearly display the graphic.

Adding a Background

To add a background graphic or replace an existing background graphic of a submap, follow these steps. The map must be opened with read-write access.
1. In the **Edit** pull-down menu, click **Select Background Picture**.
   The graphical interface displays the Submap Description dialog box for the open submap.

2. Complete or change the Background Graphics field by entering a line with the complete path and name of the graphic file. For example, to select the background graphic for a map of the United States, enter the following:
   
   /usr/OV/backgrounds/usa.gif

   Or, click **Select** and choose a graphic from the list that is displayed.

3. To display the new background graphic, click **OK**.

   If you do not select a background graphic, the background plane remains empty.

### Removing a Background

To remove a background graphic, delete the input file name from the Background Graphics field. The full path name must be deleted. The map must be opened with read-write access.

1. In the **Edit** pull-down menu, click **Select Background Picture**.
   The Submap Description dialog box for the open submap is displayed.

2. Select the Background Graphic field to make it the active field. Make sure the cursor is at the beginning of the text.

3. **Press Ctrl** and then **Delete**.
   The graphic name is removed and the field is blank.

4. Click **OK** to close the Submap Description dialog box.

### Arranging Symbols

You can arrange the symbols that are displayed on the submaps either automatically or manually. Whichever method you choose, you must have a read-write map to save the new layout. You can also change the symbol labels. This section provides the steps for arranging and labeling symbols.

### Using Redo Layout

Redo layout enables you to arrange the symbols in a submap according to the assigned layout algorithm. Redo layout works regardless of whether automatic layout is enabled or disabled.

To save the new layout for the submap, you must have read-write access. New symbol positions are not saved for read-only maps.

#### Steps

To redo the layout for a submap, follow these steps:

1. In the **View** pull-down menu, click **Redo Layout**.

2. If you redo the layout for a read-only map, the graphical interface displays a warning message that the new symbol positions will not be saved. If you redo the layout for a read-write map, the graphical interface displays a warning message that all symbols will be repositioned and their current positions cannot be restored.

3. Click **OK** to redo the map. The graphical interface repositions the symbols on the submap according to the submap layout algorithm.
Setting Automatic Layout

Automatic layout enables the system to automatically arrange the symbols in a submap according to the assigned layout algorithm. You can turn automatic layout on or off for the current submap, or for all submaps. You must have the map open with read-write access.

Steps
To set automatic layout, follow these steps:
1. In the View pull-down menu, click Automatic Layout.
2. The graphical interface displays a cascade menu. You can turn automatic layout on or off for the current submap or for all submaps.
3. After you make your selection, the graphical interface applies the change.

Note: If you turn automatic layout off, for either the current submap or for all submaps, the graphical interface requires a holding area to place the newly discovered objects. The graphical interface displays an area at the bottom of the current submap or all submaps called the New Object Holding Area. The New Object Holding Area is displayed only when there are symbols to be placed.

To manually move an object from the New Object Holding Area, select the object using the Ctrl button and mouse button 2 and drag it to the current submap.

To enable the graphical interface to move all objects from the New Object Holding Area to the current map, click Redo Layout in the View pull-down menu or turn automatic layout on.

Modifying and Displaying Symbol Labels

If you have changed the name of a resource and you want to change the label for the symbol, follow the procedure to modify and display symbol labels.

Steps
To modify symbol labels, follow these steps. You must have the map open with read-write access.
1. Click the symbol of the object with mouse button 3 to display the context menu.
2. In the context menu, click Edit -> Modify/Describe -> Symbol.
3. Enter the text in the Label field on the Symbol Description dialog box.
4. Click Yes or No to the right of Display Label to select whether to display the symbol label in the submap.
5. In the Symbol Description dialog box, click OK to apply the change and close the dialog box.

Note: If there are too many symbols on a submap, the labels are not displayed.

In the Edit pull-down menu, click Show/Hide Labels to control the display of selected symbols.

If you want to change the symbol on a submap, click Change Symbol Type in the Edit context menu.
Assigning Maps

You can provide read-write access for each user by creating a unique map for each user. The map can be a copy of the map named default. Users can make changes to their own map without effecting what is displayed on other maps. When you create a map, you become the owner and the only one with read-write access to the map. Use the `ovwchown` command to change the owner of the map from you to the user.

Setting Map Permissions

To prevent maps from being deleted, set up different map permissions on the various map databases.

Use the `ovwperms` command or the Configure --> Change Maps owner/group/mode option available through the Server Setup application to set permissions on maps. Setting map permissions to read-only for users prevents them from deleting specific maps. Only the root user can then delete a map. You must be a root user to change map permissions.

See "Assigning Map Access Access Levels" on page 56 for information about the different types of map access.

The `ovwperms` command changes the permissions for all files and directories associated with the map. See the ovwperms man page for more information about changing map permissions with a command.

Steps

To change map permissions using the Server Setup application, follow these steps:

1. Enter `serversetup` on the command line to access the Server Setup application.
2. Click Configure --> Change Map(s) owner/group/mode.
   A dialog box is displayed.
3. Enter the necessary information in the fields.
4. Click OK.

If you have a number of users that share the same maps and applications or perform similar tasks, use the UNIX `mkgroup` command to create a group for them. Groups can be formed for users that share access authority to protected resources. After your group is established, you can change map permissions for the group rather than for individuals.

Customizing the Menu Bar and Tool Palette

If you are not using Tivoli NetView security services, which provides a customized graphical interface based on Tivoli NetView group permissions, you can use the OVwRegDir environment variable to customize each user's menu bar and tool palette with the applications that pertain to only that user.

See Chapter 2, "Defining and Managing a Security Policy", on page 21 for information about Tivoli NetView security services. The OVwRegDir environment variable points to the location of the application registration files. You can set this variable in the user's .profile or .kshrc file. Placing the desired registration files in each user's directory gives each user access to a different set of operations. You can
point to individual directories then, from those directories, set up symbolic links to the registration files you want in /usr/OV/registration/C.

If there are specific menu items you do not want users to access, edit the ovw application registration file to remove those entries. However, doing so prevents anyone from using those options because they no longer exist.

An alternative to deleting menu items is to split the ovw application registration file. For example, you can split the file in two, ovw1 and ovw2, and separate the options. Add the OVwRegDir environment variable to the user’s .profile to point to either the ovw1 or ovw2 registration file.

For example, let’s say we split the ovw file into ovw1 and ovw2. You have two users, Lou and Judy. If you want Lou to have access to the menu items in ovw1 and Judy to have access to the menu items in ovw2, add the following line to each of their .profiles:

```bash
export OVwRegDir=/u/lou/reg/C
For Lou

export OVwRegDir=/u/judy/reg/C
For Judy
```

Copy ovw1 to Lou’s directory:

/u/lou/reg/C/ovw1

Copy ovw2 to Judy’s directory:

/u/judy/reg/C/ovw2

Before editing the ovw registration file, make a copy of the original for a backup file.

### Changing the Graphical Interface Defaults

There are certain graphical interface characteristics that you might want to change. For example, you might want the navigation tree and tool palette to display as icons, or you might not want them to be active at all when the Tivoli NetView product is started. The resources and defaults for these characteristics are defined in the /usr/OV/app-defaults directory for all users.

You can change a default for each user by copying the line containing the resource you want to change into the users $HOME/.Xdefaults file. Changing the resource in the app-defaults files affects all users. You might prefer to put the entries in the .Xdefaults file because customized settings in the app-defaults files are overwritten if you apply a service fix, but the settings in the .Xdefaults file are not. Settings in the .Xdefaults file override the settings in the app-defaults files.

If you change any of the app-defaults files after the application you are customizing is started, such as Tivoli NetView or the Event Display application, use the command `xrdb -merge .Xdefaults` to load the new resource and restart the application. If you have not started the application you are customizing, the new resource will be loaded when you start the application.

To change the files in the /usr/OV/app-defaults directory, you must be a root user.
Window Resources

Resources are found in the /usr/OV/app-defaults/OVw file, which defines the resources for the NetView Windows server. Although there are too many to list them all here, if you browse the file, you will find resources for the following:

**Fonts**  You can change all fonts in the graphical interface, such as button fonts, label fonts, and window title fonts.

**Colors**  You can change all colors in the graphical interface, such as background colors, symbol colors, and connection colors.

**Sizes**  You can change the default sizes for all windows in the graphical interface, such as message windows, submap windows, and the tools window.

Other Resources

Table 13 lists files, resources, and defaults found in the /usr/OV/app-defaults directory. This is not a complete list. Browse the files to see a complete list. These are resources that affect the appearance of applications displayed in the graphical interface. If you are changing the resources for an individual, you must have read-write access to the .Xdefaults file in the $HOME directory you are changing. If the .Xdefaults file does not exist, you can create this file.

<table>
<thead>
<tr>
<th>What You Want Changed</th>
<th>File</th>
<th>Resource</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event card format presentation</td>
<td>Nvevents</td>
<td>nvevents.initialPresCard</td>
<td>True</td>
</tr>
<tr>
<td>Event cards text color</td>
<td>Nvevents</td>
<td>nvevents<em>card</em>cardTextColor</td>
<td>black</td>
</tr>
<tr>
<td>Event cards color</td>
<td>Nvevents</td>
<td>nvevents.cardColor</td>
<td>#fdecfd32d2</td>
</tr>
<tr>
<td>Event application text color (text that does not appear on the cards)</td>
<td>Nvevents</td>
<td>nvevents*foreground</td>
<td>black</td>
</tr>
<tr>
<td>Number of events in a workspace</td>
<td>Nvevents</td>
<td>nvevents.maxLoadEvents</td>
<td>500</td>
</tr>
<tr>
<td>Number of workspaces per session</td>
<td>Nvevents</td>
<td>nvevents.maxNumWS</td>
<td>500</td>
</tr>
<tr>
<td>Save event workspaces</td>
<td>Nvevents</td>
<td>saveEnvOnExit</td>
<td>False</td>
</tr>
<tr>
<td>Start event application with saved workspaces</td>
<td>Nvevents</td>
<td>loadEnvOnInit</td>
<td>False</td>
</tr>
<tr>
<td>Include static workspaces and workspaces that were loaded using the File --&gt; Load option for saving</td>
<td>Nvevents</td>
<td>considerStaticWrkSpcs</td>
<td>False</td>
</tr>
<tr>
<td>Start main window and control desk as icon</td>
<td>OVw</td>
<td>OVw*shellIconify</td>
<td>False</td>
</tr>
<tr>
<td>Start the tool palette as icon</td>
<td>OVw</td>
<td>OVw*toolShellIconify</td>
<td>False</td>
</tr>
<tr>
<td>Start the navigation tree as icon</td>
<td>OVw</td>
<td>OVw*navTreeShellIconify</td>
<td>False</td>
</tr>
<tr>
<td>Start the navigation tree</td>
<td>OVw</td>
<td>OVw*navTreePresent</td>
<td>True</td>
</tr>
<tr>
<td>Start the tool palette</td>
<td>OVw</td>
<td>OVw*toolPalettePresent</td>
<td>True</td>
</tr>
<tr>
<td>Start control desk with events minimized</td>
<td>OVw</td>
<td>OVw*thereAreInternalTools</td>
<td>True</td>
</tr>
<tr>
<td>Start control desk with network view</td>
<td>OVw</td>
<td>OVw*controlDeskHasBox</td>
<td>True</td>
</tr>
<tr>
<td>Graph line width</td>
<td>XNm</td>
<td>xnmgraph.lineWidth</td>
<td>2</td>
</tr>
<tr>
<td>Graph line colors</td>
<td>XNm</td>
<td>xnmgraph.graphLineColors</td>
<td>See file</td>
</tr>
</tbody>
</table>
Steps for Changing the Graphical Interface Defaults

To customize the graphical interface defaults for a user ID, follow these steps:

1. Examine the files in the /usr/OV/app-defaults directory to determine which items you want to customize.

2. Edit the .Xdefaults file in the user’s $HOME directory and implement your changes. For example, if you want to change the default font for the Tree button on a submap window, enter a line with the following format:

   OVw*treeButtonFont:  

   Where fontname is the name of the font that you want. For more information about fonts, colors, and sizes, refer to the system documentation.

3. After you change the .Xdefaults file, save it. To see the changes, you must exit the Tivoli NetView product and restart it or use the command xrdb -merge .Xdefaults to load the new resource.

Preventing the Control Desk from Automatically Starting

By default, the Control Desk is active when you start the graphical interface. If you do not want the Control Desk to start, edit the /usr/OV/registration/C/ovsnmp/nvevents file, and remove the –Initial flag from the following line:

Command -Shared -Initial "$[nvevents:-/usr/OV/bin/nvevents]" ;

Note that the Control Desk will start by running any application, such as the Event Display application, that uses the Control Desk, and you can start the Control Desk from the tool palette. When you apply Tivoli NetView service, a new /usr/OV/registration/C/ovsnmp/nvevents might replace your customized file, and you might need to remove the -Initial flag from the new file.

Customizing the Failing Resource Display

If you have a read-write map you can display all the failing resources for an object by clicking Tools —> Failing Resources Display. This option opens a new submap with symbols that represent the failing resources of a selected object. This new submap is not connected to the submap hierarchy for the currently opened map. The failing resource submap starts its own hierarchy which you can see in the Navigation Tree.

The failing resource symbols reflect the current status of all the interfaces of an object. If an interface status changes, the symbol representing that interface also changes. When you double-click on a failed resource, another submap displays the hierarchy of the failed resource within the network.

You can customize the Failing Resource Display by editing the /usr/OV/registration/C/xnmfault file and setting the following flags:

- **t**  Changes the text in the title bar.

- **r**  Changes the capability field so that only certain objects are displayed. For example, you might want to display only those objects that have the field isConnector set to True.

- **s**  Changes what the status of the resources must be to be displayed. For example, you can change the status so that only marginal status is displayed.

- **u 011**  A zero (0) indicates that if symbols matching the criteria are not found,
continue checking and display a submap when one is found. A one (1) removes the symbols if the status does not match the criteria.

For changes to take effect, restart ovw. The following is an example of the command line in the xnmfault file:
Command "/usr/OV/bin/xnmfault -t 'title' -r isConnector -s 2 "$OVw Selectionl";

**Customizing Event Filters for Users**

You can start the nvevents application with activated filters. During startup, the nvevents application reads an event filter file located in the user’s $HOME directory or in the directory indicated by the resource profileDir in the /usr/OV/app-defaults/Nvevents file. The filter file is named after the user with the extension of events. For example, if the user name is shannon, and you did not modify the .Xdefaults file, the filters are defined in $HOME/.shannon.events.

You can define which filters you want activated for each user. The events file name and the filter rule name must be defined in the user’s events file using the following syntax:

FilterFileName filterFileRule

For example, on the AIX operating system, if you create an events file to activate filters Trap_to_Alert_Threshold and Receive_from_6611_router, the contents of the events file will look like the following example:

/usr/OV/filters/filter.samples Trap_to_Alert_Threshold
/usr/OV/filters/filter.samples Receive_from_6611_router

Where filter.samples is the filter file name and Trap_to_Alert_Threshold is the filter file rule.

When you activate filters during the nvevents operation, they are saved in this file. The next time the application is started the last filter(s) activated will be automatically registered. See Chapter 5, “Correlating, Filtering, and Configuring Events”, on page 109 for more information about filtering events.

**Customizing the Tivoli NetView Grapher**

Many of the Tivoli NetView product’s graph applications present their results in graphs that you can save, print, or customize to better suit your needs. Although each application graphs different information, the general presentation format is the same.

Graph applications have certain display default values. These defaults might be suitable for your purposes, but there might be other times when you need to modify the default display to meet your needs. This section describes the ways you can customize the graph display.

Figure 20 on page 103 shows the graphical interface’s standard layout for graph applications, using the Network Activity —> Interface Traffic option in the Monitor pull-down menu.
This time, the application was started from the Tivoli NetView main menu, so it is displayed in the control desk. You might choose to start it from the Tools window and not put it in the control desk.

**Entering Numeric Values**

The following rules apply when entering numeric values in dialog box fields:

- A number beginning with x or 0x is treated as hexadecimal. For example, 0x0010 becomes decimal 6.
- Any other number beginning with 0 is treated as octal. For example, 010 becomes decimal 8.
- Any number beginning with a numeric digit other than 0 is treated as decimal. For example, 10 becomes decimal 10.
- If you enter a value containing a digit that is not valid for its format, the number is truncated at the place that was not valid. For example, 123A56 becomes 123.

**Setting Time Intervals**

In the View pull-down menu, click **Time Intervals** to see the time statistics associated with the graph application. The Display Interval at the top of the dialog box shows the date and time the application was started and is dynamically updated with the current date and time. Move the slider box below the Display Interval to change the interval for which graphed data is displayed.

By default, the graph display shows the most recent 5 minutes of results. The vertical bars on the display mark the minutes. You can change the display width by typing a new interval in the Display Width text field and clicking the **Apply** button. Notice that a beginning date and time and an ending date and time are displayed directly above the Display Width text field.
You can change the resolution of the graph by clicking the Resolution by data/Resolution user defined option button. Resolution by data means that the graph displays the data exactly as it is collected. If you click Resolution user defined, you can increase the resolution, which normalizes the data and enables you to see trends. Refer to the Help system for more information about changing the graph resolution.

The SNMP Polling On / SNMP Polling Off option button enables you to set the frequency with which a graph of real-time data is updated. This button has no effect on historical data. The update frequency determines how often the device is queried and any new data displayed in the graph.

**Changing the Line Configuration**

You can change the characteristics of the lines on the graph. In the View pull-down menu, click **Line Configuration** to display the Line Configuration dialog box, as shown in Figure 21.

The Line Configuration dialog box displays a row of fields for each line on the graph. The fields of the Line Configuration dialog box are described in the following list.

**Data Label**

This section lists the names of all MIB objects for which the application is collecting and graphing information. By clicking a name from the list that is displayed when you click the option button, you can choose which MIB objects you want to display. This feature is useful when an application gathers data on more MIB objects than the maximum number that can be displayed at one time in the graph.

**On/Off**

If you have many lines on your graph and want to simplify the
presentation, turn some lines off by clicking the On/Off toggle button. You can also turn lines off by clicking the lines on the graph with the Ctrl button and mouse button 2. The lines’ data labels are also removed from the display. To turn the lines back on, click the On/Off toggle button to On.

**Color**
You can change the color of each line by selecting a color from the list that is displayed when you click the Color toggle buttons. If it is difficult for you to distinguish certain colors and, therefore, differentiate the lines, you can display the label for the line by clicking the line on the graph with mouse button 2.

**Line Width**
You can change the line width by clicking a line width from the selections that appear when you click the Line Width toggle button.

**Multipliers**
Sometimes the collected data includes values that are reported in different units of measurement. You can display all of them on one graph for comparison by changing multipliers, so they are all based on the same unit of measurement. The default is to display data with no multiplier.

**Displayed Values**
By default, the average values are displayed. You can select any of the following values for display:
- Minimum
- Average
- Minimum and Average
- Maximum
- Minimum and Maximum
- Average and Maximum
- Minimum, Average, and Maximum

For example, if you select Minimum and Average, the graph will display two lines for the same MIB object, one that reflects the Minimum value and one reflecting the Average value.

**Getting Application Statistics**
In the View pull-down menu, click Statistics in the graph application to see traffic statistics about each line on the graph. Information about the minimum, average, maximum, and last values is displayed in a table where each line of graphed data forms a row of the table. The statistics are updated based on the value specified for SNMP polling intervals. These statistics are used for analyzing trends about performance peaks and valleys.

By default, the raw values are shown. You can select a multiplier to change the y-axis increments. Doing so also changes the shape of the lines on the graph. In the View pull-down menu, select Line Configuration to change the multiplier. Choose a value from the list that is displayed when you click the Multiplier option button.

**Checking Application Messages**
You can look at all the messages that have accumulated for the graph application by clicking Messages in the View pull-down menu. The Messages dialog box displays error and informational messages associated with the application’s processing. Output from the File —> Memory Usage and File —> Line Info operations is also stored in this dialog box.
Paging through the Graph

If you are collecting data over a long period of time, the graph cannot display all of it at once. In the View pull-down menu, click Screen Paging for a Help dialog box that explains how to view different parts of the collected data. You can page backward or forward, one screen at a time, or center the graph around a selected point in time.

Scaling the Y-Axis

You can scale the y-axis of the graph in one of the following ways:

- On all data
- On displayed data

The default display scales the y-axis on displayed data. You can use the default if you are not running the application for a long period of time. However, if you run the application for a long time, there might be a greater fluctuation from high to low values, which might not be reflected in the portion of the graph currently being displayed. If you need to base subsequent decisions on the overall pattern of data, consider changing the scaling.

Changing the Display from Color to Monochrome

The default graph presentation uses a different color for each line that represents a monitored MIB object. The colors help you track which line goes with which MIB object.

You can change to monochrome mode and still differentiate each line on the graph. In the View pull-down menu, click Color/Monochrome and then click Monochrome. Monochrome mode uses different types of lines, for example, solid or dotted, to represent each MIB object’s values. You might want to use this option to see how the graph would look if it were to be printed on a monochrome printer. You can use the Print Tool to see the results.

Displaying or Hiding the Grid

By default, a grid is displayed for all graph applications. You can choose to hide the grid by clicking Show/Hide Grid in the View pull-down menu. The x-axis and y-axis remain, but the vertical lines that mark each minute disappear, as do the horizontal lines that extend the numerical divisions on the y-axis across the width of the graph.

Showing Counter Values

If you are graphing MIB values of type Counter, you can click Show counters As in View pull-down menu and choose one of the following ways to display the data:

- Rate of Change
  The default value, which shows the new counter value as a time-averaged value since the last query for the MIB object.

- Actual Sampled Value
  Shows the actual value returned from the MIB Counter variable.

- Delta Value
  Shows the actual change in the MIB variable since the last query for the MIB object. This value is not time-averaged.
For example, suppose you are graphing a MIB variable with the following statistics:

- Value of MIB variable at time 0 --> 100
- Value of MIB variable at time 10 --> 300

The value from time 10 would be graphed in the following ways:

**Rate of Change**
- 20 (derived from (300−100)/10)

**Actual Sampled Value**
- 300

**Delta Value**
- 200 (derived from (300−100))

**Note:** MIB expressions containing counter values will not be affected by this setting. These MIB expression values will still be represented as a rate of change.

### Adding a Line

An application sometimes has more lines to graph than the maximum number that are specified for the graph when the application is created. In the View pull-down menu, click Add Line to temporarily increase the number of lines the graph can display. This selection will be unavailable (grayed out) if the number of lines to be graphed exceeds the maximum number of lines that can be graphed, as defined in the application's app-defaults file.

### Using the Context Menu

The context menu in a graph application enables you to zoom in and out so you can look at the graph from different perspectives. You can also use the context menu to page forward or backward through the collected data, or to display the beginning of the data, the end of the data, or all of the data.

### Printing Graphs

You can use the Print Tool application to print graphs. See “Printing Graphed Data” on page 214 for these steps.
Chapter 5. Correlating, Filtering, and Configuring Events

To help you manage a network effectively, the Tivoli NetView product must receive information about changes that affect objects in the network. **Events** generated by agents that monitor network objects convey this information to the Tivoli NetView product.

Large networks with many objects and agents can generate so many events that the manager is flooded with traffic and must devote an excessive amount of time to processing incoming events. In addition, the manager generates events when it polls agents for the status of network objects. **Event correlation rules** and **event filters** can help you control the amount of:

- Event traffic to be displayed on the Tivoli NetView graphical interface
- Traps that are forwarded to the Tivoli Enterprise Console product
- Traps that are forwarded to the Tivoli NetView for z/OS product for further handling

This chapter contains the following topics:
- “Events: General Information” on page 109
- “Starting the Event Display Application” on page 118
- “Viewing the Event Log” on page 121
- “Correlating Events” on page 122
- “Creating Event Filters” on page 142
- “Activating and Deactivating Event Filters” on page 148
- “Configuring Events” on page 153
- “Displaying a Warning Window for Events” on page 158
- “Converting Events to Alerts (AIX Only)” on page 159
- “Sending Alerts to the Host Program (AIX Only)” on page 160

Events: General Information

The Tivoli NetView product uses the following types of events:

**Map events**
Notifications issued because a user or application does something that affects the status of the current map or of the Tivoli NetView graphical interface. For example, if you add a connection between a workstation and a server on a submap, an event is generated and logged in the event log file. The contents of the submap change to include the added connection.

**Network events**
A message sent by an agent to one or more managers to provide notification of an occurrence affecting a network object. These events are not necessarily reflected in the map. For example, if an SNMP agent is not in your management region, but is configured to send traps to the manager, you will receive events for that agent.

Logging Events

All of the events and SNMP traps received by the Tivoli NetView product are logged in the /usr/OV/log/ovevent.log file by default. From this file, the nvevents application reads the events filtered for display and displays them through the **Event History application**.
See “Viewing the Event Log” on page 121 for information about using the Event History application to view events stored in this file.

SNMP traps can also be logged in the /usr/OV/log/trapd.log file. This file is in ASCII format, so you can edit it to view logged traps, or print the contents of this file. If the Tivoli NetView product is configured to work with a relational database, you can transfer trapd.log data to a relational database and use the relational database tools to create reports.

Refer to the Tivoli NetView for UNIX Database Guide for more information about transferring trapd.log data to a relational database.

**Information Provided by SNMP Traps**

SNMP defines six generic types of traps and allows definition of enterprise-specific traps. The trap structure conveys the following information to the Tivoli NetView product:

- Agent’s object that was affected
- IP address of the agent that sent the trap
- Event description (either a generic trap or enterprise-specific trap, including trap number)
- Time stamp
- Optional enterprise-specific trap identification
- List of variables describing the trap

The agent knows which manager system to send traps to by use of a user-configurable trap destination. The manager system can then retrieve more information to isolate a problem by polling the agent system.

**Information Provided by Tivoli NetView Internal Events**

The Tivoli NetView product internally generates events, which are treated as enterprise-specific traps. These events include the following information:

- Description of the event.
- Name of the node associated with the event in the system name (sysName) MIB variable.
  
  A node name with the value <none> refers to the manager station running the Tivoli NetView product.

To look at a list of the Tivoli NetView product’s internally generated events, enter the [usr/OV/bin/event -l] command at the command line. See Appendix A, “Tivoli NetView Internal Traps”, on page 281 for detailed information about the internal traps.

**Forwarding Events to the Tivoli Enterprise Console**

The Tivoli NetView product provides a default set of significant events that are forwarded to the Tivoli Enterprise Console including status events, selected SNMP data collection threshold events, and Router Fault Isolation events. See Table 35 on page 315 for a list of events that are forwarded by default.

The TEC_ITS.rs event correlation rule is used to forward this default set of events. The corresponding Tivoli Enterprise Console rule set, which provides special event correlation for the TEC_ITS events, is active by default for IBM Tivoli Enterprise Console, Version 3.8 and later.
The following scripts are provided to configure event forwarding:

<table>
<thead>
<tr>
<th>Script</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>instalnv</td>
<td>Use to configure event forwarding during a new installation</td>
</tr>
<tr>
<td>/usr/OV/bin/nvits_config</td>
<td>Use to configure event forwarding after a new installation using the Tivoli Framework</td>
</tr>
<tr>
<td>/usr/OV/bin/tecits_upgrade</td>
<td>Use to upgrade to the new TEC_ITS event class structure at any time after a Tivoli NetView upgrade installation</td>
</tr>
</tbody>
</table>

The TEC_ITS event class structure was introduced in Tivoli NetView, Version 7.1.2. All classes in the TEC_ITS event class structure begin with the prefix TEC_ITS instead of OV. The event class structure was modified, including some slot mappings, for Tivoli NetView events that are sent to the Tivoli Enterprise Console. See Table 36 on page 316 for mapping of Tivoli NetView events to the new TEC_ITS classes that are not forwarded by default.

Notes:

1. If you upgrade from a previous version of the Tivoli NetView product, you can continue to use the old OV class structure to preserve their custom rules. However, you will not be able to use the enhanced integration capabilities provided by the new TEC_ITS event class structure and rules. You can choose to upgrade at any time after an upgrade installation using the tecits_upgrade script. See “Using the tecits_upgrade Script to Upgrade to TEC_ITS Event Class Structure” on page 113 for more information.
2. If you are forwarding events from the Tivoli NetView product to a version of the Tivoli Enterprise Console product that is earlier than version 3.7, you must set the Pre37Server flag to YES in the /usr/OV/conf/tecint.conf file by adding the entry Pre37Server=YES. You must stop and restart the nserverd daemon after adding this flag. Use either the ovstop and ovstart commands or Server Setup to stop and restart the daemon.

You can also configure the Tivoli NetView product to forward events to the Tivoli Enterprise Console using the Server Setup application as follows:

1. Enter serversetup on the command line to access the Server Setup application.
2. Click Configure —> Configure Event Forwarding to IBM Tivoli Enterprise Console. The Configure Event Forwarding to the IBM Tivoli Enterprise Console window is displayed.
3. Click the button next to Forward events to IBM Tivoli Enterprise Console server?. Yes is displayed on the button.
4. Type the host name of the system on which the Tivoli Enterprise Console server is installed.
5. Type the server port number. The default value is 0 (zero) for a UNIX TEC server and 5529 for a Windows TEC server.
6. Either type the Tivoli NetView correlation rule TEC_ITS.rs, or click Select to display a list of rule names.
7. If you want to use the Tivoli Management Environment transport mechanism, click the button next to Use TME® transport mechanism. Yes is displayed on the button.
8. If you are using the Tivoli Management Environment transport mechanism, type the instance number of the Tivoli Endpoint that points to the Tivoli
Management Environment server on which the Tivoli Enterprise Console server is installed. Click **Select** to display a list of Tivoli Endpoints.

9. Click **OK**.

The `nvserverd` daemon forwards to the Tivoli Event Server only the events that meet the specified event correlation rule. See "Correlating Events" on page 122 for information about event correlation rules.

### Format of Events Forwarded to the Tivoli Enterprise Console

Event information is formatted as a set of attributes, or slots. Each attribute is a predefined information slot that contains the attribute name and attribute value. For each event, the event adapter does the following:

- Maps the event to a Tivoli Enterprise Console event class
- Maps the event variable bindings to slots in the class
- Sends this information to the Tivoli Event Server, which subsequently forwards the event information to the Tivoli Enterprise Console product.

The default event class for these events is TEC_ITS. The Tivoli NetView BAROC and rules files are shipped with the Tivoli Enterprise Console product.

The default event slot mappings are described in the following list:

<table>
<thead>
<tr>
<th>source</th>
<th>nvserverd</th>
</tr>
</thead>
<tbody>
<tr>
<td>sub_source</td>
<td>Tivoli NetView source character corresponding to the internal component of the Tivoli NetView product that generated the event</td>
</tr>
<tr>
<td>origin</td>
<td>IP address of trap origin</td>
</tr>
<tr>
<td>sub_origin</td>
<td>&lt;empty&gt;</td>
</tr>
<tr>
<td>hostname</td>
<td>Host name of the resource being reported on</td>
</tr>
<tr>
<td>msg</td>
<td>Text description of the trap</td>
</tr>
<tr>
<td>date</td>
<td>Date the Tivoli NetView product received the event</td>
</tr>
<tr>
<td>severity</td>
<td>Configured severity for the trap</td>
</tr>
<tr>
<td>status</td>
<td>OPEN</td>
</tr>
<tr>
<td>nv_enterprise</td>
<td>Enterprise of the trap</td>
</tr>
<tr>
<td>nv_generic</td>
<td>Generic trap number</td>
</tr>
<tr>
<td>nv_specific</td>
<td>Specific trap number</td>
</tr>
<tr>
<td>nvhostname</td>
<td>Hostname or IP address of the Tivoli NetView server</td>
</tr>
<tr>
<td>fqhostname</td>
<td>Fully qualified hostname of the resource being reported on</td>
</tr>
<tr>
<td>nv_var1 through nv_var15</td>
<td>Variable bindings 1 through 15; the remaining variable bindings are not forwarded.</td>
</tr>
</tbody>
</table>

You can change the event class and event slot mappings using the Event Configuration application. See "Configuring Events" on page 153 for more information.

**Note:** Any traps containing variable bindings that might have values equal to any of the following keywords will fail to parse correctly by the Tivoli Enterprise Console event parser. Do not set `nv_var12=INTEGER`. Because the `nv_var1` through `nv_var15` default to their corresponding variable binding, any variable binding that could possibly contain a keyword should not match an
event attribute. For example, a MLMThresholdArm trap might contain INTEGER in variable binding 12. This, by default, maps to nv_var12. Because this might cause a parsing error in Tivoli Enterprise Console, map nv_var12 to something else.

Reserved Tivoli Enterprise Console keywords are as follows:
- DEBUG
- DEFINES
- END
- ENUMERATION
- INTEGER
- INT32
- ISA
- I_NAME
- LIST_OF
- POINTER
- REAL
- REFERENCES_TO
- SELF
- SINGLE
- default
- dup_detect
- parse
- print_ref
- reverse
- self_classname

For more information, consult the Tivoli Event Integration Facility User’s Guide.

### Controlling the Event Adapter

When the Tivoli NetView product is configured to forward events to the Tivoli Enterprise Console product, the event adapter is automatically started. If you want to stop forwarding events to the console, use the `/usr/OV/bin/nvtecia -stop` command.

**Note:** If you restart the Tivoli NetView product, the event adapter automatically restarts. To permanently stop forwarding events to the console, use the Server Setup Application to reconfigure the Tivoli NetView product.

If you modify the ruleset you are using to forward events to the Tivoli Enterprise Console, use the `nvtecia -reload` command to reload the ruleset.

Refer to the nvtecia man page for more information.

### Using the tecits_upgrade Script to Upgrade to TEC_ITS Event Class Structure

During an upgrade installation of the Tivoli NetView product, the old OV class structure and all user customizations for the Tivoli Enterprise Console product are preserved. You can continue to use your custom rules with the old class structure.

At any time after an upgrade installation, you can use the `tecits_upgrade` script to upgrade to the new TEC_ITS event class structure. You will need to reconfigure your customizations after performing this upgrade.

Usage:
/usr/OV/bin/tecits_upgrade [-s serverName] [-p port] [-w]
/usr/OV/bin/tecits_upgrade -T[-e endpoint]
/usr/OV/bin/tecits_upgrade -h

Where:

-e The endpoint instance to use when using the TME communication method.
   This parameter is optional. The default is 1.
-h Displays the usage statement.
-p port Specifies the port to be used for communication with the Tivoli Enterprise
   Console server. This parameter is optional. It is necessary to specify this
   parameter only if the port has been customized, otherwise the default
   (5529 for a Microsoft Windows Tivoli Enterprise Console server, 0 for a
   UNIX Tivoli Enterprise Console server) is used. This parameter is not
   necessary unless you have never configured event forwarding to a Tivoli
   Enterprise Console server.
-s serverName The name of the Tivoli Enterprise Console server to which events are being
   forwarded. This parameter is not necessary unless you have never
   configured event forwarding to a Tivoli Enterprise Console server.
-T Specifies that the TME communication method be used for the Tivoli
   Enterprise Console server.
-w Indicates that the Tivoli Enterprise Console server specified with the -s
   option is a Windows machine. This option must be specified for a
   Windows Tivoli Enterprise Console server, otherwise the port will not be
   configured correctly.
   This parameter is not necessary unless you have never configured event
   forwarding to a Tivoli Enterprise Console server.

The tecits_upgrade script does the following:
- Backs up the current trapd.conf file.
- Upgrades traps to the new TEC_ITS class structure including new slot mappings
  for some events.
- Updates the tecint.conf file (if it exists) to use the new TEC_ITS ruleset and class
  structure. If the tecint.conf file does not exist, and you have not specified the
  Tivoli Enterprise Console server information with the -s (and possibly -w and
  -p) options, you are prompted for the Tivoli Enterprise Console server name and
  related information.
- Sets the NVMAPGLOBALACK environment variable to 1. This is required for
  the Acknowledge function to work from the Tivoli Enterprise Console. It will
  change the default behavior of Acknowledge to use global-based mode. When
  global-based mode is set, the Acknowledged status is an object attribute. All
  NetView clients are notified when an object is acknowledged or
  unacknowledged immediately, regardless of which map is open or if the action
  is performed on a read-only map.
- Creates a new CiscoDevices SmartSet.
- Adds the Bandwidth Utilization for Routers and CPU Utilization for Cisco
  Devices SNMP data collections.
Enabling Tivoli Enterprise Console Event Severity Escalation for Nodes

The Tivoli NetView program provides the ability to send network fault (availability) information to the Tivoli Enterprise Console. The Tivoli NetView program can also determine which applications or services are running on an end point. This information is also sent to the Tivoli Enterprise Console, which can then use this information to present the correlated network and application events to the operator. Operators can then prioritize events based on the impact to an application. See the Tivoli Enterprise Console Rule Builders Guide for information about how the Tivoli Enterprise Console uses these service impact events to escalate severity and correlate it to events.

The Tivoli NetView program discovers applications or services using servmon discovery tasks and IBM Tivoli Monitoring product integration.

Servmon Discovery Tasks
The servmon daemon works closely with the network discovery daemon to discover applications running on endpoints or nodes. Servmon provides several different discovery tasks, including tasks to discover the DB2 application, the WebSphere Application Server, and WebSphere for MQSeries®. These tasks are extensible and customizable. Note that these tasks must be customized for installations that do not use the default ports for these products.

See “Discovering Services on Nodes” on page 176 for more information.

IBM Tivoli Monitoring Integration
The Tivoli NetView program can communicate with user-configured IBM Tivoli Monitoring servers to determine the list of endpoints that are being monitored and which applications are being monitored on each end point. This information is stored in the object database for use in determining the service impact to network faults. See “Forwarding Service Event Information to the Tivoli Enterprise Console” for more information.

Use the ITMQUERY function to query IBM Tivoli Monitoring servers for information about IBM Tivoli Monitoring server endpoints and the services that are installed on these endpoints. See “ITMQUERY Function” on page 116 for more information.

The IBM Tivoli Monitoring servers must be defined to the Tivoli NetView program. See “ITMQUERY Configuration files” on page 116 for information about discovering additional services and see the --add-server parameter description in “ITMQUERY Command Parameter Descriptions” on page 116.

Forwarding Service Event Information to the Tivoli Enterprise Console
The NetView Tivoli Enterprise Console adapter forwards service-related events for IBM Tivoli Monitoring nodes to the Tivoli Enterprise Console to enable correlation and escalation of the following events:

- TEC_ITS_NODE_STATUS or TEC_ITS_ROUTER_STATUS event with a status of either DOWN or MARGINAL:
  A TEC_ITS_NODE_SERVICE_IMPACT event is generated for each service attribute that is defined for the node. Each of these events provides the host name and the affected service name for the node.

- TEC_ITS_SUBNET_CONNECTIVITY event with a status of UNREACHABLE:
A TEC_ITS_SUBNET_SERVICE_IMPACT event is generated for each service that is found within the subnet. Each of these events provides the service name (for example IBM_WebSphere_MQ) and the names of all nodes within the subnet that contain the service.

**ITMQUERY Function**

Use the ITMQUERY function to perform the following tasks:

- Add, delete, or modify IBM Tivoli Monitoring server information that is defined in the itm_servers.conf configuration file.
- List information about IBM Tivoli Monitoring endpoints.
- List information about services that are installed on IBM Tivoli Monitoring endpoints.
- Define your own log to use for IBM Tivoli Monitoring information.
- Create netmon seed files. See “Creating netmon Seed Files” on page 117 for more information.

**ITMQUERY Configuration files:** This section provides information about the following configuration files that are used by ITMQUERY function:

/usr/OV/conf/itm_servers.conf

This file lists the IBM Tivoli Monitoring servers that you want to monitor and the account information for each server. The information is used by the ITMQUERY function, and by the service monitor function of servmon to perform IBM Tivoli Monitoring attribute discovery tests.

Do not use an editor to change this file. Instead, use either the procedure described in “Using Server Setup to Enable the ITMQUERY Function” on page 118 or use the ITMQUERY --add-server or --remove-server parameters to modify this file.

/usr/OV/conf/itm_attributes.conf

This file is used to specify the names of services that the ITMQUERY function should search for when it queries IBM Tivoli Monitoring endpoints to determine which services are installed on the endpoints. The servmon daemon also uses this file for service discovery purposes.

Use a text editor to modify this file. See the /usr/ov/conf/itm_attributes.conf file for more information about the syntax of the file entries, and for information about adding and removing entries.

**ITMQUERY Command Parameter Descriptions:** This section describes the ITMQUERY command parameters.

Where:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Displays the ITMQUERY man page.</td>
</tr>
<tr>
<td>--add-server servername</td>
<td>Adds server servername to file itm_servers.conf. You are prompted to provide the IBM Tivoli Monitoring user ID and password.</td>
</tr>
<tr>
<td>--remove-server servername</td>
<td>Removes server servername from the itm_servers.conf file.</td>
</tr>
<tr>
<td>--verify-server-info</td>
<td>Verifies that information for each server defined in the itm_servers.conf file is valid. An attempt is made to log on to each server that is listed to</td>
</tr>
</tbody>
</table>
verify that the server name, user ID, and password are valid. The following is an example in which one server has been configured:

```
itmquery --verify-server-info
```

Attempting to verify connection information for all configured IBM Tivoli Monitoring Servers

```
nvvm01.tivlab.raleigh.ibm.com ... Succeeded
```

--dump-endpoints Displays the IP addresses of all endpoints being monitored by the IBM Tivoli Monitoring servers listed in the itm_servers.conf file.

```
--dump-products-for-endpoints true | false
```

If set to true, the recognized products for each endpoint that is listed by the --dump-endpoints parameter is displayed. If set to false, a list of endpoint IP addresses is displayed. The default value is true.

--server When using the --dump-endpoints parameter, you can use this option to specify the single server to use when performing the query.

```
--logconfig
```

Defines an alternate log file to use instead of the default /usr/OV/conf/itmquery-log4j.properties log file.

Creating netmon Seed Files: You can use the ITMQUERY function to obtain IP addresses that can be used to create a netmon seed file. For example, you can create a netmon seed file that contains:

- The IP addresses of all the endpoints that are being monitored by one or more IBM Tivoli Monitor servers.
- The IBM Tivoli Monitor endpoints that have the WebSphere product installed for all IBM Tivoli Monitor servers that are defined in file.

For example, you could enter the `/usr/ov/bin/itmquery -dump-endpoints` command and use the following output to manually add one or more of these endpoints to a netmon.seed file:

```
# Endpoints for IBM Tivoli Monitoring Server itm1.ibm.com
10.107.2.111 # (WAS, DB2)
10.107.2.113 # (WAS)
10.107.2.155 # (MQ)
```

Function: You can configure a list of IBM Tivoli Monitoring servers for Tivoli NetView to query using one of the following methods:

- During a new installation using the instalnv script, you are prompted for this information if you specify the `--server` option to configure event forwarding to the Tivoli Enterprise Console product. The Enter Server Connection Information for IBM Tivoli NetView window is displayed.
- Specify the `--server` option when you run the nvits_config script. The script displays the Enter Server Connection Information for IBM Tivoli NetView window.
- Use the Server Setup application, `/usr/ov/bin/serversetup`. Click **Configure** —> **Configure Connection to IBM Tivoli Monitoring**. A command window is displayed and then the Enter Server Connection Information for IBM Tivoli NetView window is displayed.

See “Using Server Setup to Enable the ITMQUERY Function” on page 118
Using Server Setup to Enable the ITMQUERYFunction: Use the following procedure to configure a list of IBM Tivoli Monitoring servers for the Tivoli NetView product to query using the Server Setup application:

1. Enter `serversetup` on the command line to access the Server Setup application.
2. Click **Configure —> Configure Connection to IBM Tivoli Monitoring**. The Enter Server Connection Information for IBM Tivoli NetView window is displayed.
3. Click **Add**. The IBM Tivoli Monitoring Server window is displayed.
4. Type the host name or IP address of the server.
5. Type the user ID for the server.
6. Type the password.
7. Click **OK**. The Tivoli NetView product verifies the information. If the information is incorrect, you can do one of the following:
   - Click **Save** to save the changes and proceed.
   - Click **Discard Changes** to continue without saving the changes.
   - Click **Retry** to return to the IBM Tivoli Monitoring Server window and change the information.
8. Repeat this procedure to add more servers.
9. Click **OK** when you are finished

---

**Starting the Event Display Application**

The Event Display application can be started in any of the following ways:

- The Event Display application starts automatically when the Tivoli NetView product starts. It displays all events received during the current Tivoli NetView session that have been filtered for display. Only one instance of this application can be open per session to display all the filtered events of that session. However, you can open other instances to display the filtered events of that session for selected network objects.

- You can also select events for display in a separate workspace window. Events are synchronized in client machines. For example, if an event is cleared, notes are added, or the event severity and category is changed in a workspace on a client machine, the change is reflected in the workspaces on all the clients.

- Select a symbol on a submap. In the **Monitor pull-down menu**, click **Events —> Current Events**. In this case, the application displays all uncleared events (events currently displayed in the main workspace window) associated with the selected symbol.

- Click mouse button 3 on a symbol in a submap to display the object context menu and click **Monitor —> Events —> Current Events**. The Event Display application displays all uncleared events associated with the selected symbol.

- Select an object or objects on the submap and drag the Events icon from the Tools window to the Control Desk window or another area of the desktop. Events for each object will be displayed in separate dynamic workspaces.

You can start the Tivoli NetView product without starting the Event Display application. If you are a root user, you can edit the `/usr/OV/registration/C/ovsnmp/nvevents` file and remove the **-Initial** flag from the command that initiates the Event Display application. Although you make this change, you can still start the Event Display application from the main menu, an object context menu, or the Tools window.
When Workspaces Are Automatically Created

The Tivoli NetView product automatically creates a workspace in the following circumstances:

- The first time the Event Display application is started, the workspace contains the dynamic event display for the current Tivoli NetView session. It is located in the Control Desk window. You can also start this application by dragging its icon from the Tools window.
- Each time you select a node from the submap and then click **Events —> Current Events** in the **Monitor** pull-down menu or drag the Events icon from the Tools window, a new workspace is opened for the node you selected.
- The first time you click **Events —> Event History** in the **Monitor** pull-down menu a new workspace that contains events from the `/usr/OV/log/ovevent.log` file is opened instead of events being received during the current Tivoli NetView session. See “Viewing the Event Log” on page 121 for more information about the event history application.
- When you drag an event card from either the Event Display or Event History applications.
- Each time you click **Dynamic Workspace** in the **Create** pull-down menu in the main workspace window.

Refer to the **Tivoli NetView for UNIX User’s Guide for Beginners** for information about creating a dynamic workspace.

Saving the Event Workspace Configuration

You can change the following resources in the `/usr/OV/app-defaults/Nvevents` file so that you can save your event workspace configuration and start the Event Display application with the saved configuration:

**saveEnvOnExt**

Defines whether the Event Display application saves all workspaces. The default value is False. When you change the value for `saveEnvOnExt` to True, the current event workspace configuration is saved in a configuration file in the `$USER/$HOME/NvEnvironment` directory when you exit the Event Display application.

**loadEnvOnInit**

Defines if the Event Display application starts with a saved event workspace configuration. The default value is False. When you change the value for `loadEnvOnInit` to True, all saved workspaces are opened inside the Control Desk when you start the Event Display application.

**considerStaticWrkSpcs**

Defines if the Event Display application saves and loads static workspaces in addition to dynamic workspaces. When you change the value for `considerStaticWrkSpcs` to True, static workspaces and workspaces that were loaded using the **File —> Load** option are saved and opened when you start the Event Display application. The default value is False, which saves and loads only dynamic workspaces. Save and load the configuration for dynamic workspaces to avoid starting the Event Display application with saved static workspaces which can cause a delay in loading the event workspace configuration.

You can override the value set for `saveEnvOnExt` by clicking the **Save Environment** in the **Options** pull-down menu in the Event Display application main workspace. When the value for `saveEnvOnExt` is True, a radio button is displayed next to the **Options —> Save Environment** option indicating that this
option is active and the event workspace configuration will be saved when you exit the Event Display application. In the Options pull-down menu, click Save Environment to deactivate the option, and the current event workspace configuration will not be saved. If the value for loadEnvOnInit is True, the Event Display application will start with the previously saved configuration, if one exists.

Similarly, when the value for saveEnvOnExt is False, a radio button is not displayed before the Options —> Save Environment option indicating that this option is not active and the event workspace configuration will not be saved when you exit the Event Display application. Click the Options —> Save Environment option to activate the option and save the current event workspace configuration.

Whether the configuration is saved because the value for saveEnvOnExit is True or because you clicked the Save Environment in the Options pull-down menu, the Event Display application is not started with the saved workspace configuration unless the value for loadEnvOnInit is True.

Controlling How Events Are Displayed

You can change the following resources in the /usr/OV/app-defaults/Nvevents file, which control how events are displayed:

workListDetailMode
Controls how events are displayed when events are in list format. The valid values are:

0 Double-clicking an event opens the card in a static workspace. This is the default.
1 Double-clicking an event opens the card in a static workspace that is resizable.

workCardDetailMode
Controls how events are displayed when events are in card format. The valid values are:

0 Double-clicking a card brings the card to the top of the stack. This is the default.
1 Double-clicking a card opens the card in a static workspace.
2 Double-clicking a card opens the card in a static workspace that is resizable.

Changing the values for the workListDetailMode and workCardDetailMode resources to 1 and 2, respectively, enables you view all the information on the card. You can resize the workspace window to resize the card in the same proportion as the window.

Suppressing Events from Unmanaged Nodes

To suppress events from IP nodes that are unmanaged in the open map, click Suppress Traps from Unmanaged Nodes in the Options pull-down menu in the Event Display window. This menu option is a toggle button. To resume seeing the traps, select this menu option again.

Searching for Events

You can search for events based on predefined filtering criteria or on any or all of the following criteria:

- A search string
• An event source
• An event category
• An event severity level

Searching by Criteria
To search for events by criteria, follow these steps:
1. In the Search pull-down menu in the Event Display Application, click By Criteria.
2. Enter the search criteria in the Search String field on the Search by Criteria dialog box. You can also search for a specific event source, category, or severity.
3. If you want the search results displayed in a separate workspace, click Create Workspace. Otherwise, the events are selected in the Event Display Application Main Workspace. Selected event cards are a darker shade than the others.
4. Click OK to start the search and close the dialog box. Events meeting the search criteria are either selected or displayed in a Static Workspace.

Searching by Filter
To search for events by filtering criteria, follow these steps:
1. In the Search pull-down menu in the Events Display Application, click By Filter.
2. Select the name of the filter from the list of Available Filters in File. You can click File List to enter changes to the list of available filters. You can also edit the selected filter by clicking Display/Edit.
3. Select Activate to activate the filter. The dialog box is closed. A workspace is created in the Control Desk window containing the filtered events.

Viewing the Event Log
The Event History application, which displays events from the /usr/OV/log/ovevent.log file, starts in one of the following ways:
• When you click Events —> Event History from the Monitor pull-down menu.
• When you drag the Event History icon from the Tools window and drop it either in the Control Desk window or another location on the desktop.

The displayed events come from the /usr/OV/log/ovevent.log file, which contains events from the current Tivoli NetView session. The number of events that can be stored in this file at one time, and potentially displayed by the event history application, depends on the maximum allowable size of the file.

See “Changing the Size of the Event Log” on page 122 for more information.

When the Event History application is started, events are not displayed until you click Display Events in the Query pull-down menu. This delay in displaying events gives you the opportunity to apply filter criteria to the query of the /usr/OV/log/ovevent.log file, so the display does not contain an unmanageable number of events. If there are too many objects selected for inclusion in the filter, the filter will not be activated, and the following message will be displayed: Sieve creation failed. Error in Object creation.

In the Options pull-down menu, click Filter Control to filter the Event History display. From the Filter Control dialog box, you can activate and deactivate filters to control the event history display. You can also use the Filter Editor to create a filter if the list of available filters does not contain one that suits your needs.
See “Using the Filter Editor” on page 143 for more information.

Refer to the online help for information about the other Event History menu options.

Changing the Size of the Event Log

To change the size of the /usr/OV/log/ovevent.log file, click **Options —› Set Log Size** in the Event History menu. The default log size is 128 KB. The maximum log size is 2 MB. Changing the default log size through the Event History menu changes the size for the current editing session only. You can configure the ovelmd daemon to change the size permanently through the Tivoli desktop.

Refer to *Tivoli NetView for UNIX Configuration Guide* for those steps. The following conditions affect changing the size of the log file:

- If you specify a new size that is less than the current size of the file, the current file becomes the backup log file and a new log file is started. You might do this if you want to clear the current log file and record only events from a given time in the current session.
- If you specify a new size that is greater than the current size of the log file, events continue to be added to the current file.
- If you set the size to zero (0), event logging is turned off until you enter a positive integer. Events already logged will not be deleted unless the log file is deleted.

Correlating Events

You can create a ruleset that correlates or compares incoming events to event processing decisions and actions. The ruleset editor enables you to graphically create a rule comprised of event-processing decisions and actions that are represented by icons (nodes).

You can create rulesets to:

- Enhance the filtering capabilities for dynamic event displays. You can filter for MIB variables inside traps and define thresholds based on event data.
- Use object database fields to correlate events.
- Automatically remove resolved events from the Event Display application.
- Override severity and object status associated with an event.
- Automatically issue a call to a pager.

Processing Events within a Ruleset

The following list describes how the decision and action nodes process an event through a ruleset:

**Decision**

If the decision is determined to be true, the event is passed on to the next node (or nodes if there are multiple connections) in the ruleset. If the decision is determined to be false, processing halts on that path through the ruleset. When all processing of an event halts and the event is not passed on to the next node, the event is deleted.

**Action**

The action is passed to the **actionsvr** daemon, which starts a new process for the specified action, and the event is passed on to the next node in the ruleset. If the action does not complete successfully, the ACTF_EV
(59179071) action failed trap is generated. All actions requested and the events which caused those actions are logged. Actions include:

- Forwarding the event to the Event Display application
- Overriding the severity or object status associated with the event
- Resolving the event
- Issuing any operating system command, script file or executable, or Tivoli NetView command
- Issuing a call to a pager
- Setting a global variable to some value
- Setting a Tivoli NetView object database field
- Setting a MIB variable

Configuring the Paging Utility

To use the Tivoli NetView paging utility through an event correlation rule (using the Pager node) or from the command line (using the `nvpage` command) with an analog line for modem communications, perform the following steps for the modem that is attached to your system:

1. Add and configure a tty device for modem communications using the UNIX `mkdev` command.
2. Test the modem communication through the tty device using a communications program, such as `ate`, provided with your operating system.
3. Customize the paging utility configuration files, which are located in the `/usr/OV/conf` directory:
   - `nv.carriers`
     Lists the defined carriers. Add the appropriate entries for all paging carriers used at your site. The Numeric IDs accepted on Modem line: Y/N field indicates the pager type. If numeric IDs are accepted, the pager type is numeric. If numeric IDs are not accepted, the pager type is alpha.
     See the `nv.carriers` man page for more information.
   - `*.modem`
     Contains the default information for the modem. The asterisk (*) represents the name of the modem file. The following modem files are provided:
     - `ibm5853.modem`
       For the 2400 baud IBM Model 5853 modem
     - `ibm7855.modem`
       For the IBM Model 7855
     - `newhayes.modem`
       For most Hayes compatible modems
     - `oldhayes.modem`
       For Hayes compatible modems that do not understand the extended AT command set
     - `qblazer.modem`
       For Hayes compatible modems
     - `blank.modem`
       For you to copy and customize
Usually, you do not need to change the values in the modem file. If a modem file is not provided for the modem you are using, use the blank.modem file as a template.

See the \texttt{modem} man page for more information.

\textbf{nvpager.config}

Lists the defaults that are the physical characteristics of the modem. Specify the tty device that you already configured and tested and change the modem characteristics to reflect the values configured on the tty device. Add the name of the modem file that corresponds to the modem dedicated to paging. See the \texttt{nvpager.config} man page for more information.

\textbf{nvpaging.protocols}

Defines the characteristics of the following paging protocols: TAP, IXO, PET, and PAKNET. The Protocol field in the nv.carriers file specifies the paging protocol being used and points to the nvpaging.protocols file for configuration information. If you are using a paging protocol similar to TAP, IXO, PET, and PAKNET, copy the information provided for one of these protocols and modify it with the appropriate information for the protocol you are using.

See the \texttt{nvpaging.protocols} man page for more information.

After updating the configuration files, stop and restart the \texttt{nvpagerd} daemon to make the changes available to the paging utility.

4. Create Tivoli NetView security user profiles for those individuals who you want to page automatically through an event correlation ruleset. See "Creating and Changing a User Profile" on page 31 for those steps.

When you use the \texttt{Pager node} in an event correlation ruleset, you specify the user ID of the person you want to page. The Tivoli NetView security user profile defines the user ID and the paging information. It is not necessary to activate security to access the paging information in a user's profile.

You can also send a page from the command line using the \texttt{nvpage} command. See the man page for more information.

\textbf{Types of Ruleset Nodes}

The ruleset editor contains the following nodes as shown in Figure 22 on page 136:

\textbf{Action}

Specifies the action to be performed when an event is forwarded to this node. Fields from the trap being processed are available as environment variables. The specified action can be any operating system command, the full path name of any shell script or executable, or any Tivoli NetView command. Usually, the output from the specified action is displayed on the screen. If the output is not displayed on the screen, it is written in the \texttt{/usr/OV/log/nvaction.log} file. You can use this node to execute the \texttt{/usr/OV/bin/ovxecho} command to display a dialog window when a specific event occurs.

The dialog box contains one relevant field: Action. Enter any operating system command, the full path name of any shell script or executable, or any Tivoli NetView command.

\textbf{Block event display}

Prevents events from being forwarded to the Event Display application.
Use this node if you have changed the default processing action to pass (forward) events to the Event Display application and you do not want to forward events that meet specific conditions. A trap that is processed through this node is marked so that it will not be handled by the default processing action specified for the ruleset.

The dialog box does not contain any relevant fields.

**Check Route**
Checks for communication between two network nodes and forwards the event based on the availability of this communication. For example, you can use this node to check the path from the manager to a device before forwarding a node down trap.

*Note:* The check route node does not check the status of the node; it checks only the availability of the path to the node.

The dialog box contains the following relevant fields:

**Source**
Specifies the node where the check route starts.

**Destination**
Specifies the node for which you are checking the route.

**Forward Event when**
Specifies forwarding of the event to the next node if the path is available (communication is successful) or unavailable (communication fails).

**SNMP Defaults**
Specifies the following SNMP defaults: community name, number of retries, remote port, and local port.

**Compare MIB Variable**
Compares the current value of a MIB variable against a specified value. When a trap is processed by this node, the ruleset processor issues an SNMP GET request for the specified MIB variable.

The dialog box contains the following relevant fields:

**MIB Variable Name**
Specifies the fully-qualified name, including the instance number, of the variable you want to compare.

**Object ID Source**
Specifies the trap attribute to be used to determine the object from which to get the specified MIB variable.

**Community Name**
Specifies the community name to be used in the SNMP request.

**Value to Compare**
Specifies a literal value that will be compared to the MIB variable value. This value must correspond with the type of MIB variable. If the type of the MIB variable is integer, for example, this value must be numeric.

**Comparison Type**
Specifies the type of comparison to be made.
You can use the **Browse MIB** button to start the MIB browser and then cut and paste information from the MIB browser into the Compare MIB Variable dialog box.

**Event Attributes**
Comparing any attribute of the incoming event to a literal value. You can use this node to check for events generated by a particular device.

The dialog box contains the following relevant fields:

- **Attribute**
  - Specifies the name of the attribute to be compared

- **Comparison Type**
  - Specifies the type of comparison to be performed

- **Value**
  - Specifies the literal value to be compared to the specified trap attribute, such as a host name

See [“Event Attribute Values” on page 134](#) for more information.

**Forward**
Forwards the event to applications that have registered to receive the output of the ruleset. A trap that is processed through this node is marked so that it will not be handled by the default processing action specified for this rule.

The dialog box does not contain any relevant fields.

**Inline Action**
Specifies the action to be performed when an event is forwarded to this node. Unlike a command specified in an Action node, a command specified in an Inline Action node is not sent to the actionsvr daemon. Instead, the command is executed immediately, and processing continues to the next node if the return code of the action matches the return code you specify within the specified time period.

The dialog box contains the following relevant fields:

- **Command**
  - Specifies any operating system command, the full path name of any shell script or executable, or any Tivoli NetView command.

- **Wait Interval**
  - Specifies the time period, in seconds, that the ruleset processor should wait for the specified action to return. Values can be in the range of 0–999 seconds. If the wait interval is 0, the return code from the action is ignored and processing immediately proceeds to the next node. If a wait interval is specified, and the return code from the action is not received in the wait interval, it is considered to be a failure and processing does not proceed to the next node. If the action is not completed within the specified time period, processing will not proceed to the next node.

- **Command exit code comparison**
  - Specifies the type of comparison you want to make.

- **Exit Code**
  - Specifies the return code value from the specified action that you want to use in the comparison.

- **Override**
  - Overrides the object status or severity assigned to a specific event and
updates applications that have registered to receive the output of the ruleset. The Event Display application is registered to receive the output. For example, you can use this node to change the severity to Major when a node down event is received for a router. Use this node with the Query Database Field node to override status or severity for specific device types.

The dialog box contains the following relevant fields:

**Status**
Specifies the new object status to be associated with this event or click **no override** if you do not want to change the status. The Event Display application updates the object status to this value.

**Severity**
Specifies the new severity level to be used for this event or click **no override** if you do not want to change the severity level.

A trap that is processed through this node is marked so that it will not be handled by the default processing action specified for this rule.

**Pager**
Issues a call to a pager that has been defined in a Tivoli NetView user profile. You should have already configured the paging utility. See “Configuring the Paging Utility” on page 123 for those steps.

The paging utility will use the pager number and carrier information defined in the user profile.

The dialog box contains the following relevant fields:

**User ID**
Specifies the Tivoli NetView user ID of the person to be paged. If pager information is not found in the Tivoli NetView user profile or there is no Tivoli NetView ID for the user, a dialog box is displayed in which you can enter the User ID and pager information. Then a user profile is created or updated.

**Message Text**
Specifies message text to be delivered with the page. The message can include trap data passed in environment variables. See “Environment Variables for Trap Data” on page 135 for more information.

**Pass on Match**
Compares some attribute of the event being processed with an attribute of all traps received in a specified period of time.

You can specify one or a maximum of ten attribute comparisons. The Attribute List section contains the list of attribute comparisons that you want to evaluate. When you define an attribute comparison in the dialog box and click **Add** the attribute comparison is added to the list. The attribute comparisons will be examined in the order in which they are listed in the Attribute List. If the first pair of attributes matches, then the second pair of attributes is examined, and so on through the list. Click **Reorder** to change the order in which the attribute comparisons are evaluated.

You can stop processing when the first incoming event matches the criteria you defined, or you can continue processing to find matches in all incoming events received in the specified time period. Specifying multiple attribute comparisons on multiple events can be used with the Resolve node to resolve all interface down or node down events for a specific device when an interface up or node up event is received for that device.
When the attribute specified in Event 1 Attribute, in the current trap, matches the attribute specified in Resetting Event Attribute for all attribute comparisons in the Attribute List, in any trap received during the specified time period, processing continues to the next defined node.

The dialog box contains the following relevant fields:

**Event 1 Attribute**
Specifies the name of the attribute in the first event.

**Comparison Type**
Specifies the type of comparison to be performed.

**Resetting Event Attribute**
Specifies the name of the attribute in the second event. Click the **Add** button to add the attribute comparison to the Attribute List.

**Event Retention**
Specifies the length of time the first event will be held to wait for the second event. The maximum event retention value is 999 hours, 59 minutes, 59 seconds. If an attribute match is found between two events within the specified period of time, processing continues to the next node in the ruleset.

**Match on multiple events**
Controls whether processing stops when the first incoming event received in the specified time period matches all the criteria in the Attribute List or whether processing continues to find matches in all incoming events received in the specified time.

To modify an attribute comparison, follow these steps:

1. Select the attribute comparison you want to change in the Attribute List. The comparison values are entered in the Event 1 Attribute, Comparison Type, and Resetting Event Attribute fields.
2. Change the fields as necessary and click **Modify**. The comparison values in the Attribute List are updated.

To change the order in which the attribute comparisons are evaluated, select the comparisons you want to move in the Attribute List and click the up or down arrows on **Reorder** to move the comparisons up or down one position.

To delete an attribute comparison, select the appropriate attribute comparison in the Attribute List and click **Delete**.

See “Event Attribute Values” on page 134 for more information.

**Query Database SmartSet**
Tests whether the node is a member of the specified SmartSet.

The dialog box contains the following relevant fields:

**Object Source ID**
Specifies the source of the object ID to be used in querying the database.

**SmartSet Name**
Specifies the name of the SmartSet to be tested. Click on the **Select** button to display a list of all currently defined SmartSets. You can
also specify a name of a SmartSet that does not yet exist but you intend to create. See “Defining and Managing SmartSets” on page 83 for more information.

**Forward event when**
Specifies forwarding of the event to the next node if the node is contained in the specified SmartSet or if the node is not contained in the specified SmartSet.

**Query Database Field**
Compares a value from the Tivoli NetView object database to a literal value or to a value contained in the incoming event. You can use this node to check if the originating device is a router.

The dialog box contains the following relevant fields:

**Field Name**
Specifies the name of the database field to be queried. Enter the name of the database field or click the **Select**.

**Object ID Source**
Enter the fully-qualified object name of the device whose field is to be queried or click **Select**.

**Comparison Type**
Specifies the type of comparison to be performed.

**Compare Field to**
Specifies either a literal value or an attribute value to be used in the comparison.

**Reset on Match**
Compares some attribute of the event being processed with an attribute of all traps received in a specified period of time. This node is similar to the **Pass on Match node** except that if a match is found, the event is not passed on to the next node in the ruleset and processing stops.

You can specify one or a maximum of ten attribute comparisons. The Attribute List section contains the list of attribute comparisons that you want to evaluate. When you define an attribute comparison in the dialog box and click **Add**, the attribute comparison is added to the list. The attribute comparisons will be examined in the order in which they are listed in the Attribute List. If the first pair of attributes matches, the second pair of attributes is examined, and so on through the list. Click **Reorder** to change the order in which the attribute comparisons are evaluated.

You can stop processing when the first incoming event matches the criteria you defined, or you can continue processing to find matches in all incoming events received in the specified time period. Specifying multiple attribute comparisons on multiple events can be used to check for all interface down or node down events for a specific device when an interface up or node up event is received for that device. When the attribute specified in Event 1 Attribute, in the current trap, matches the attribute specified in Resetting Event Attribute for all attribute comparisons in the Attribute List, in any trap received during the specified time period, processing through the ruleset stops.

You can use this node to discard events before they are forwarded to the Event Display application. You might find this node useful for events that are generated from a device that frequently goes up and down. The dialog box contains the following relevant fields:
Event 1 Attribute
Specifies the name of the attribute in the first event.

Comparison Type
Specifies the type of comparison to be performed.

Resetting Event Attribute
Specifies the name of the attribute in the second event.

Delay Time
Specifies the length of time the first event will be held to wait for the second event. The maximum event retention value is 999 hours, 59 minutes, 59 seconds. Click Add to add the attribute comparison to the Attribute List.

If an attribute match is found between two events for all attribute comparisons in the list within the specified period of time, the event is not forwarded to the next node in the ruleset and processing stops.

Match on multiple events
Controls whether processing stops when the first incoming event received in the specified time period matches all the criteria in the Attribute List or whether processing continues to find matches in all incoming events received in the specified time.

To modify an attribute comparison, follow these steps:
1. Select the attribute comparison you want to change in the Attribute List. The comparison values are entered in the Event 1 Attribute, Comparison Type, and Resetting Event Attribute fields.
2. Make the necessary changes to these fields and click Modify. The comparison values in the Attribute List are updated.

To change the order in which the attribute comparisons are evaluated, select the comparisons you want to move in the Attribute List and click the Reorder button to move the comparisons up or down one position.

To delete an attribute comparison, select the appropriate attribute comparison in the Attribute List and click Delete.

See “Event Attribute Values” on page 134 for more information.

Set Database Field
Sets the value of any Tivoli NetView non-Boolean object database field. Fields that have TRUE or FALSE values cannot be changed.

The dialog box contains the following relevant fields:

Field Name
Specifies the name of the field in the object database that you want to change

Object ID Source
Specifies the source of the object ID to be used in selecting an object from the object database

Set Value to
Specifies either a literal value or an event attribute value to be used for the database field setting
Query Global Variable
Queries the value of the global variable that has been previously set using the Set Global Variable node.

The dialog box contains the following relevant fields:

Variable Name
Specifies the name of the variable you are checking. The variable is created and assigned a value using the set global variable node.

Comparison Type
Specifies the type of comparison to be performed.

Compare Variable to
Specifies either a literal value or an event attribute value to be used in the comparison.

See “Event Attribute Values” on page 134 for more information.

Set Global Variable
Sets a variable for use within the ruleset. For example, use this node to set a flag whose value will be checked later in the ruleset using the Query Global Variable node. When the ruleset is finished processing, the global variable is no longer in effect.

The dialog box contains the following relevant fields:

Variable Name
Specifies a user-defined text string associated with the value of the variable, such as flag.

Set Variable
Specifies one of the following settings:

- Increment Value by One
  If the global value has already been set, the value will be increased by one. If the global variable has not yet been set, the value will be set to one.

- Decrement Value by One
  If the global value has already been set, the value will be decreased by one. If the value has not yet been set, the value will be set at negative one.

- Set to Literal Value
  Enter a text value in this field.

- Set to Attribute Value
  Click Select to select a trap attribute.

See “Event Attribute Values” on page 134 for more information.

Set MIB Variable
Issues an SNMP SET command to set the value of a variable in the MIB representing any network resource. For example, you can use this node to change the system contact for a particular device.

The dialog box contains the following relevant fields:

MIB Variable Name
Specifies the fully-qualified name, including the instance number, of the variable you want to change.
Variable Data Type
Specifies the type of data to be placed in the MIB field, such as integer, string, and so on.

Object ID Source
Specifies where to get the object ID whose MIB is to be changed.

Community Name
Specifies the community name of the object whose MIB is to be changed.

Value to be Set
Specifies a literal value to be used as the data for the MIB field value. The data type of this value must match the type specified in the Variable Data Type field.

See “Event Attribute Values” on page 134 for more information.

Click Browse MIB to start the MIB browser and then cut and paste information from the MIB browser into the Set MIB Variable dialog box.

Resolve
Forwards a message to all registered applications indicating that a previous event has been resolved. By default, the Event Display application is registered to receive the output from rulesets. The receiving application determines how to handle a trap that has been forwarded from this node. This node is frequently used in conjunction with the Pass on Match node. You can use the Resolve node to delete an interface or node down event from the Event Display application when an interface or node up event is received. A trap that is processed through this node is marked so that it will not be handled by the default processing action specified for the ruleset.

The dialog box does not contain any relevant fields.

Set State
Sets the correlation state of an object in the Tivoli NetView object database. The current state is updated in the corrstat1 field in the object database, and the previous value in the corrstat1 field is moved to the corrstat2 field. This process continues until the current state and as many as four previous states are stored in the object database. You can view the correlation state by selecting the object and then selecting the Display Correlation Status option from the context menu.

The dialog box contains the following relevant fields:

State Value
Specifies the text string that you want to store in the corrstat1 field of the specified object

Object ID Source
Specifies the source of the object ID to be used in selecting the object in the object database

See “Event Attribute Values” on page 134 for more information.

Thresholds
Checks for repeated occurrences of the same trap or of traps with one or more attributes in common. You can use this node to forward an event after receiving the specific number of the same event received within a
specific time period. Use this node with the Trap Settings node to identify a specific trap number. The dialog box contains the following relevant fields:

**Type**
Specifies when the event should be forwarded by selecting one of the following values:

- **First**
  When a threshold condition is reached, forwards the first \( n \) traps to the next node, where \( n \) is the number specified in the Count field

- **At**
  When a threshold condition is reached, forwards the \( n \)th trap to the next node, where \( n \) is the number specified in the Count field

- **After**
  When a threshold condition is reached, forwards all traps after the \( n \)th trap to the next node, where \( n \) is the number specified in the Count field

**Count**
Specifies the number of traps required to reach the threshold condition

**Time Period**
Specifies the length of time within which the number of events specified in the Count field must be received to reach the threshold condition. Use this field in conjunction with the Time Unit field.

**Time Unit**
Specifies the unit of measure (minutes, seconds, hours, or days) for the number specified in the Time Period field.

**Threshold by Attribute 1—Threshold by Attribute 9**
By default, the threshold depends on the trap ID. If you want to define on one or more trap attributes, select the Threshold by Attribute 1 button and then click on the **Select...** button to specify the first attribute in the trap to be used. Use the scroll bar in the Threshold by Attribute panel to define up to nine trap attributes. For each attribute you want to define, click the appropriate Threshold by Attribute button and then select the trap attribute to be used.

**Threshold by Attribute 1**
Specifies the first attribute in the trap to be used. By default, the threshold depends on the trap ID. Select Threshold by Attribute 1 and then click **Select** to choose the first attribute to be used.

**Threshold by Attribute 2**
Specifies the second attribute in the trap to be used. Select Threshold by Attribute 2 and then click **Select** to choose the second attribute to be used.

**Threshold by Attribute 3**
Specifies the third attribute in the trap to be used. Select Threshold by Attribute 3 and then click **Select** to choose the third attribute to be used.

See "Event Attribute Values" on page 134 for more information.

Assume that you specify a time of 5 minutes in the Time Unit field and 10 for the number of traps in the Count field. The following list describes the use of First, At, and After in the Type field.
A burst of 20 traps is received in 5 minutes. Only the first 10 traps will be forwarded.

A burst of 10 traps is received in 1 minute. Only the 10th trap is forwarded. If traps continue to arrive at one per minute thereafter, no other traps will be forwarded because traps arriving at one per minute do not meet the threshold again. The threshold is calculated by considering the current event plus past events that are received within the same time period.

A burst of 10 traps is received in 1 minute. One trap is received every minute thereafter. Each trap received in minute 2, 3, 4, and 5 will be forwarded. Similar to above example, no more traps will be forwarded after minute 5 because traps arriving at one per minute do not meet the threshold again.

**Trap Settings**

Specifies a specific trap to be processed and is identified by a pair of generic and specific trap numbers.

The dialog box displays a list of enterprise names and IDs. When you select an enterprise ID, a list of generic and specific trap numbers for that enterprise is displayed in the Event Name and Specific fields. Select one or more traps from this list. The description of the trap you select is displayed in the Trap Description field. Click **Comparison Type** to specify the type of comparison to be performed (equal to or not equal to).

**Event Attribute Values**

Use the following event attribute values to identify the event to be processed:

- **Severity**
  Specifies the severity of the event, such as Critical.

- **Category**
  Specifies the type of event, such as a status event.

- **Source**
  Specifies the internal component of the Tivoli Netview product that generated the event, such as netmon.

- **EnterpriseID**
  Specifies the enterprise that sent the trap.

- **Origin**
  Specifies the host name generating the trap. The management station is the origin for traps generated as a result of Tivoli NetView polling operations.

- **Generic**
  Specifies the generic trap value defined by SNMP.

- **Specific**
  Specifies the specific trap value defined by SNMP.

- **sysObjectID**
  Specifies the MIB object describing the agent hardware, software, and so forth.

- **sysUpTime**
  Specifies the MIB system running time since the agent has been started.

- **Community Name**
  Specifies the community name.
1-50 Trap variable which varies by enterprise or trap.

You can use one word for a trap attribute. A word is a unit of text separated by blanks. To specify the word to be used, add a period (.) and the number of the word. For example, to use the second word of the third variable binding, specify 3.2. The ruleset processor will use the string that starts after the first blank and ends with the second blank.

Tivoli NetView internal traps use variable bindings 1 through 5 as follows:

1. Specifies the source ID and is an integer value that corresponds to the internal component of the Tivoli NetView product that generated the event, such as netmon.
2. Specifies the host name to which this trap applies.
3. Specifies the event description and is a string value containing a description of the event that was generated.
4. Specifies specific trap data and is a string value containing internal data that is specific to the type of trap that is generated.
5. Specifies the database name and must be openview.

See Appendix A, “Tivoli NetView Internal Traps”, on page 281 for more information about Tivoli NetView internal traps.

Environment Variables for Trap Data

You can specify trap data using the following environment variables:

- NVE Specifies the enterprise ID
- NVA Specifies the agent address
- NVG Specifies the generic trap number
- NVS Specifies the specific trap number
- NVT Specifies the time stamp
- NVC Specifies the community name
- NVATTR_<1-50> Specifies the MIB attribute where 1-50 is the variable binding number.

These environment variables are frequently used with the Action node and the Pager node. For example, you might include a pager message similar to the following:

Multiple authentication failures for $NVA.

Sample Rulesets

The following sample rulesets are provided:

- corrNdNu.rs
  Forwards a node down trap to nvevents and clears the event if a node up trap is received for the same device within 10 minutes.

- corrIdIu.rs
  Forwards an interface down trap to nvevents and clears the event if an interface up trap is received for the same device within 10 minutes.
See “Activating a Ruleset” on page 138 for information about how to activate a ruleset.

Creating and Editing a Ruleset

To create a ruleset, use the ruleset editor as shown in Figure 22.

You must be a root user to start the ruleset editor. To start the ruleset editor, use one of the following methods:

- In the Tools pull-down menu, click Ruleset Editor.
- Enter the `nvrsEdit ruleset_name` command at the operating system command line, where `ruleset_name` is the name of the ruleset you want to create or change.
- Double click on the Ruleset Editor icon on the tool palette.
- Drag and drop the Ruleset icon from the Tools window onto the desktop.

The ruleset editor is divided into two windows:

**Ruleset**

Contains the Event Stream icon, which represents all incoming events and is the work area for creating rulesets. The window title bar contains the name of the ruleset you are currently editing. If you started the ruleset editor without specifying a ruleset name, the window’s title bar contains the name `default.rs`. You can change the name of the ruleset when you save the ruleset or you can edit an existing ruleset by selecting the File Open option from the ruleset editor menu bar. Other menu bar operations enable you to create, modify, or delete a ruleset.
Templates
Contains the nodes that you can use to create a ruleset. Each node represents either a decision node or an action node.

Changing the Default Processing Action
You can change the default processing action, which defines what is done with a trap after it has been processed through a ruleset. The default processing action is not used for traps that have been marked by being processed through one of these nodes:
- Block event display
- Forward
- Override
- Resolve

If the trap is not processed through a Block event display, Forward, Override, or Resolve node, the trap can be passed on to interested applications or discarded. The default processing action is Block, which means that the trap is not forwarded (passed) to applications that have registered to receive the output of the ruleset, such as the Event Display application.

To change the default processing action so that events are forwarded to registering applications, double click on the Event Stream icon and click Pass. You can override forwarding specific events that have passed through a ruleset by using the Block event display node, or you can reset the default action for all events that pass through the ruleset by selecting the Block button.

Adding a Node
To add a node to a ruleset, drag and drop the appropriate node from the template area onto the work area and then connect the nodes. You can also select the appropriate node from the New pull-down menu bar option. If you maximize the work area window size, click Focus to Templates from the Edit pull-down menu in the Ruleset Editor main window to keep the Templates window in front of the work area window. The Focus to Templates option is also available from the context menu that is available in the background area of the work area. When two or more decision nodes are connected sequentially (in a straight line), the logical operator AND is used. When two or more decision nodes are connected in parallel from a single decision node, the logical operator OR is used.

When you drop a node into the work area, a dialog box is displayed that contains relevant data fields for the decision or action to be performed. Complete the dialog box fields and click OK to add the node to the ruleset. The description field in each dialog box is optional. You can use this field to document the decision or action taken at the node.

You can double-click a node at any time or select Edit from the context menu on a node to display its dialog box and view or modify the data.

Connecting Two Nodes
Connect the nodes in the ruleset to define the logic path through the ruleset. Connect decision nodes sequentially (in a straight line) to use the logical operator AND. Connect two or more decision nodes in parallel from a single decision node to use the logical operator OR.

Use one of the following methods to connect nodes:
- Click Connect Two Nodes in the Edit pull-down menu in the Ruleset Editor main window. Then select the nodes you want to connect in the work area.
• Select the node in the work area to which you want to connect another node. Then drag and drop another node into the work area. The connection will be drawn automatically.

Select the “from” node first and then the “to” node so that the event flow through the ruleset is from left to right.

**Deleting a Node**
To delete a node from a ruleset, use one of the following methods:

• Click **Delete Node** in the **Edit** pull-down menu in the Ruleset Editor main window and then select the node.

• Select the node and then click **Delete Selected** in the **Edit** pull-down menu.

• Select the node and press the **Delete** key.

• Click **Delete** from the context menu on the node.

The node and the connection to the node is deleted.

**Deleting a Connection**
To delete a connection, use one of these methods:

• Click **Delete Connection** in the **Edit** pull-down menu in the Ruleset Editor main window and then select the connection.

• Select the connection and then click **Delete Selected** in the **Edit** pull-down menu.

• Select the connection and press the **Delete** key.

• Click **Delete** from the context menu on the connection.

**Inserting Another Ruleset**
You can insert another ruleset into the ruleset you are currently editing. You might find this useful for combining rulesets or building complex rules out of existing rulesets. To insert another ruleset, follow these steps:

1. If you do not want the inserted ruleset connected to the Event Stream icon, select the node to which you want to connect the inserted ruleset. If you do not select a node, the ruleset is connected to the Event Stream icon.

2. Select **Insert** from the **File** pull-down menu. A dialog box is displayed containing a list of existing rulesets.

3. Select the name of the ruleset you want to insert and click **OK**. The ruleset is inserted into the current ruleset, and the layout is recalculated.

**Saving a Ruleset**
When you are finished editing a ruleset, save it by selecting **File —> Save** from the ruleset editor menu bar. If you want to save the ruleset with a different name than the one that is displayed in the title bar of the ruleset window, select **Save As** from the **File** pull-down menu in the Ruleset Editor main window. Then enter a name for the ruleset and click **OK**. The name of the ruleset must include a .rs extension. You might name a ruleset that resolves node down and node up traps from the save device as **resolve.rs**. Rulesets are stored in the `/usr/OV/conf/rulesets` directory.

**Activating a Ruleset**
To activate a ruleset, create a new dynamic workspace. For each dynamic workspace, you can activate only one ruleset, and you can activate one or more filters.
In the Create pull-down menu in the main workspace, click Dynamic Workspace and enter the ruleset name and any other appropriate information. The new workspace uses the ruleset and filters, if any, to determine which events are displayed. If you edit a ruleset while it is active, close and reopen the dynamic workspace window to put the changes into effect. Select the Help button on the Dynamic Workspace dialog box for information about the dialog box fields.

You can activate one or more rulesets for automatic action when you start the Tivoli NetView product by editing the /usr/OV/conf/ESE.automation file, adding the names of the rulesets on separate lines. Use this method for rulesets that perform a specific action, such as a call to a pager when a particular device goes down, and you do not want to display the events.

Attention: Do not set the default processing action as Pass in the Event Stream icon or include a Forward node in rulesets that you start from the ESE.automation file. Forwarded events are passed to the actionsvr daemon, which has no events display and no mechanism to dequeue the events. Thus, the socket connection to the nvcorr daemon will fill up and cause the nvcorr daemon and all nvevents windows to hang.

Testing a Ruleset

When you have created and activated a new ruleset, use the Diagnose —> Send event to trapd daemon option available through the Tivoli desktop to send the appropriate traps and test the results of the ruleset in the workspace you created. If you don’t have the Tivoli desktop you can use the /usr/OV/bin/event command or use Server Setup —> IBM Tivoli NetView —> Diagnose —> Send event to trap daemon.

A Threshold Example

Suppose you want to monitor a specific router. You can create a ruleset to display authentication failure events after a minimum of five events have been received from the router within one minute. You also want to display a message box indicating that multiple authentication failure events have been received from the router. Figure 23 shows how the completed ruleset looks.

![Figure 23. A Correlation Rule to Set a Threshold](image)

Here are the steps to create and activate this ruleset:
1. Click **Tools — Ruleset Editor** from the Tivoli NetView menu bar. The ruleset editor is displayed.

2. Drag and drop the Trap Settings node into the work area.

3. Complete the dialog box as follows:
   a. Select an Enterprise ID of ENTERPRISES, because this is a generic trap
   b. Select Generic Trap 4 - Authentication Failure
   c. You can add a description similar to the following to describe the purpose of this node:
      Watch for authentication failure traps.
   d. Click **OK**.

4. Connect the Trap Settings node to the event stream node.

5. Drag and drop the Event Attributes node into the work area.

6. Complete the dialog box as follow:
   a. Select **Origin** in the Attribute field.
   b. Select **Equal To** for the comparison type.
   c. Type the fully-qualified host name of the router, for example: route1.raleigh.ibm.com.
   d. You can add a description similar to the following to describe the purpose of this node:
      Watch for authentication failures from router1.
   e. Click **OK**.

7. Drag and drop the threshold node into the work area.

8. Complete the dialog box as follows:
   a. Select **After** in the Type field, enter 5 in the Count field and 1 minute in the Time field.
   b. Click **Threshold by attribute** and select **Origin** for the attribute, because you are checking for events from the same device.
   c. You can add a description similar to the following to describe the purpose of this node:
      Watch for 5 authentication failures from the router1 within 1 minute.
   d. Select the **OK** button.

9. Connect the Threshold node to the Trap Settings node.

10. Drag and drop the forward node into the work area.

11. Complete the dialog box as follows:
   a. You can add a description similar to the following to describe the purpose of this node:
      Display event after five authentication failures are received from the same device within 1 minute.
   b. Click **OK**.

12. Connect the forward node to the threshold node.

13. Drag and drop the action node into the work area.

14. Complete the dialog box as follows:
   a. In the action field, enter the following:
      ```
      /usr/OV/bin/ovxecho Multiple Authentication Failures for $NVA
      ```
   b. You can add a description similar to the following to describe the purpose of this node:
Use ovxecho to display a notice when more than 5 authentication failures are received within 1 minute.

c. Click OK.

15. Connect the action node to the threshold node.

16. Click File —> Save As to save the ruleset. Enter a name for the ruleset, such as router1.threshold.rs and click OK.

17. Activate the ruleset by creating a new dynamic workspace. In the main workspace menu bar, click Create —> Dynamic Workspace. Enter the name of the ruleset and any other appropriate information.

More Examples of Rulesets
Here are a few more examples of rulesets you might create. Each example includes the objective of the ruleset and a picture of how the completed ruleset would look.

Example 1: Create a ruleset that forwards a node down trap to the Events Display application and clears the event if a node up trap is received for the same device within 30 minutes.

Example 2: Create a ruleset that changes the system contact for all routers.

Example 3: Create a ruleset that pages the appropriate individual when an IBM router or a Cisco router goes down.
This ruleset does not display the node down traps. To activate this ruleset, add the name of the ruleset to the /usr/OV/conf/ESE.automation file.

Creating Event Filters

Many events arrive at the management station on a network. Event filters are sets of criteria that determine the following conditions:

- Which events are dynamically displayed by the Event Display application.
- Which logged events are displayed by the Event History application.
- Which events are received by applications that register to receive them.
- Which events are forwarded to a host program as alerts (AIX only). These filters are called trap-to-alert filters.

See “Activating a Trap-to-Alert Filter (AIX Only)” on page 151 for more information about using these filters.

Filtering events involves setting up criteria that an event must meet before it can be displayed or sent to another application. The data is then stored in the default directory, /usr/OV/filters, or in a directory that you create. To access filters, the Tivoli NetView product must know the location of the directory containing the filter files.

You can create more than one filter file, and each filter file can contain one or more filters. Criteria for the same event can also be placed in several filters.

Types of Filters

The Tivoli NetView product enables you to create the following types of filters:

Simple filters
Expressions that include SNMP criteria and can be standalone. They can be edited using the Simple Filter Editor.

Compound filters
Expressions that are composed of several simple filter expressions. They use nested parentheses to group simple expressions and combine them with the logical operators AND, OR, and NOT. They can be edited using the Compound Filter Editor.
When multiple simple filters are activated, the logical OR is used. That means that traps meeting either set of filter criteria will be received. Trap exclusion is more effectively implemented by combining the two simple filters using the AND operator to create one compound filter. Then, activate the compound filter.

**Accessing the Filter Editor**

You can access the filter editor in any of the following ways:

- In the **Tools** pull-down menu, click **Filter Editor**
- Enter the **filtered** command as follows:
  
  
  \[
  \text{filtered} \ -f \ <\text{filename}> \ /*\text{name of filter file}*/ \\
  \ -r \ <\text{rule}\text{name}> \ /*\text{name of filter to edit}*/ \\
  \ -e \ /*\text{edit option}; \text{default=display}*/ \\
  \]

  If you want only to display a filter, you can omit the -e option. To edit a filter, use the -e option.
- Click **Options —> Filter Control** in the Event Display application window. The Filter Control dialog box enables you to activate, deactivate, display, and edit filters.

**Using the Filter Editor**

The filter editor helps you perform the following tasks:

- Create the criteria for events you want to see.
- Organize filters in files that you specify.
- Define a threshold that enables you to modulate the number of events received over a period of time.
- Define filters that can be used independently by different applications.

When you click **Filter Editor** in the **Tools** pull-down menu, the Filter Editor dialog box is displayed, as shown in Figure 24.

![Filter Editor Dialog Box](image)

Figure 24. Filter Editor Dialog Box

The name of a filter file is displayed at the top. You can work with the filters in this file or click the **File List** button to display a list of filter files from which you
can choose. Once you have selected a filter file, the names and brief descriptions of all filters in the file are displayed in the bottom part of the window.

To see the contents of a particular filter, select it and select the **Display** button to the right of the list of filters. You might want to look at a filter to determine whether it is one that you want to use as a base for creating another filter or to combine it with other filters.

Click the **Add Simple** to display the **Simple Filter Editor** dialog box, or click **Add Compound** to display the **Compound Filter Editor** dialog box.

See “Creating Simple Filters” and “Creating Compound Filters” on page 146 for information on these filter operations.

You might decide, after looking at a filter, that you no longer need it. In that case, click **Delete** to discard the filter.

To change a filter, select it from the list and click the **Modify** button. If the selected filter can be edited using the Simple Filter Editor, that editor is displayed. Otherwise, the Compound Filter Editor is displayed.

You can organize your files on a per-application basis by storing the same filter in more than one file. Use the Copy to File operation to copy a selected filter to a different file.

**Note:** When you create a filter, make sure that you do not include more than 250 criteria. Filters used by the Event History application are limited to 40 criteria. This maximum is the sum of all choices for the following criteria:

- Enterprises selected
- Traps selected
- Objects in the IP address list
- Selection of the event logged time criteria
- Selection of frequency parameters
- Logical operators (AND, OR, and NOT)

**Creating Simple Filters**

Suppose you want to create a simple filter that sends an enterprise-specific event from the Tivoli NetView product to your events display. Follow these steps:

1. In the **Tools** pull-down menu, click **Filter Editor**.
2. In the Filter Editor window, click **Add Simple** to display the Simple Filter Editor dialog box, as shown in **Figure 25 on page 145**.
3. Complete the Filter Name and Description fields.

4. You want to receive one particular event, so select **Events Equal to Selected** in the Event Identification section of the Simple Filter Editor dialog box.

5. Click **Add/Modify** to display the Enterprise Specific Trap Selection dialog box from the list of available enterprises.

   The enterprises in this list are those that are configured in the /usr/OV/conf/C/trapd.conf file. Once you make a selection from the enterprise list, all generic and specific traps associated with that enterprise are displayed in the Available Trap Types field. The default is that all traps from the selected enterprise are included as part of the filter. However, you can modify this list.

6. If you know which enterprise-specific trap you want to filter, select it from the Available Trap Types list by clicking on it and then clicking **Select**. The selected trap is displayed in the Selected Trap Types field. If you do not know which trap you want, go back to step 5.

   If you want to add an enterprise-specific trap, enter an enterprise-specific trap number in the Specific Trap Number field and click **Add To List**. Click **OK** button to apply your changes and close the dialog box.

7. In the Object Identification section of the Simple Filter Editor dialog box, select **From Objects Equal to List**. If you selected your workstation’s symbol on a submap before you opened the Filter Editor, you can click **Add From Map** to add your workstation to the List of Objects field. Otherwise, enter either the name or the IP address of your node to the Name or IP Address field, and click the **Add to List** button.
8. You can specify time and date ranges during which this event is to be sent from your workstation to the Tivoli NetView product. For example, if you want to receive this event between 8:30 and 10:30 a.m. today, complete the Time Range section as follows:

```
<table>
<thead>
<tr>
<th>TIME RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (HH:MM:SS)</td>
</tr>
<tr>
<td>Start 08:30:00</td>
</tr>
</tbody>
</table>
```

The default date is today’s date. If you do not specify any date or time ranges, and the filter is activated, this event is displayed every time it occurs. If you only specify a time, the date defaults to today’s date, and you have to reactivate the filter each day.

9. To set a threshold that specifies how many times this event might occur before notification is sent to the Tivoli NetView product, complete the text fields in the Threshold section of the Simple Filter Editor dialog box.

For example, suppose you want 5 to be the maximum number of occurrences per minute of this event that can be generated without displaying the event in the nvevents application. Enter 5 in the Frequency field and select **Less Than or Equal To**. Enter 60 in the Time Interval (Seconds) text field.

10. Click OK to close the Simple Filter Editor dialog box.

11. If you want to look at the filter you just created, select the filter name from the list in the Filter Editor window and click Display.

12. To activate your filter for the Event Display application, select Filter Control in the Options pull-down menu in the Event Display application window. Select your filter from the list of available filters and click Activate. Now, only those events that match your filter criteria will be displayed. When activating more than one simple filter, the logical operator OR is used. If you want the criteria of several filters to be used when filtering events, use the Compound Filter Editor to combine the simple filters.

13. To select the filtered events, first click Filter in the Search pull-down menu in the Event Display application window. Next, select the filter name and click Activate. The search operation highlights all events that match the filter criteria. You can also create a separate workspace in which to store these events.

For more information about creating simple filters or the different sections of the Simple Filter Editor dialog box, refer to the Help system.

### Creating Compound Filters

Compound filters are created by joining several existing simple filters or several simple expressions with logical operators. These filters can also be used to specify CMIS expressions that cannot be specified using the Simple Filter Editor. To create a compound filter, use the Compound Filter Editor, as shown in Figure 26 on page 147.
Follow these steps:

1. In the Filter Editor window, click **Add Compound**.
2. Specify a filter name and description for the compound filter.
3. In the Filter Expression field, either type a filter expression or click **Get Filter**.
4. If you click **Get Filter**, the Get Filter Dialog box is displayed. Select a filter in one of the following ways:
   - To retrieve the name of a filter, **Get Filter Name**.
   - To retrieve the contents of a filter, **Get Filter Contents button**.

   If you need to modify the filter in any way, retrieve its contents from the Get Filter dialog box. Otherwise, if you plan to use the filter without making changes, get only its name.

   Click **OK** to close the Get Filter Dialog box. The selected filter or filter name is displayed in the Filter Expression field of the Compound Filter Editor dialog box. Now you can edit this filter expression or combine it with others.

5. Select one of the following logical operators to combine filter expressions:
   - AND, which places && at the end of the current filter expression. Combining two filter expressions with AND means that events must match all criteria of both expressions.
   - OR, which places || at the end of the current filter expression. Combining two filter expressions with OR means that events must match the criteria of at least one filter expression, but need not match the criteria of both expressions.

6. Either select another filter or enter some filter criteria in the Filter Expression field. You can combine several expressions in one filter.

7. Click **OK** to close the Compound Filter Editor dialog box.

8. If you want to save your compound filter to another filter file, click **Save as** and either select another file from the list or enter a new path name in the Selection field. Click **OK** to close the File Selection dialog box. Otherwise, click the **OK** button to close the Compound Filter Editor dialog box.

9. Click **Close** to close the Filter Editor window.
For more information about creating and using compound filters, refer to the online help.

**Activating and Deactivating Event Filters**

Filters must be activated to affect the destination and the display of events. Once a filter is activated, all events matching that filter pass through the active filter to the registered applications. Events that do not match the criteria are not permitted to pass through the filter. Therefore, filtering the events will reduce the number of events being displayed.

You can activate and deactivate filters for the Event Display application, the Event History application, a dynamic workspace, and the trap-to-alert conversion process. Being able to change the activation status of a filter gives you increased control over the number and type of events that are received and displayed. However, make sure that you understand the effect of activating or deactivating a particular filter before you make the change.

Filters can also be activated and deactivated programmatically. Refer to the *Tivoli NetView for UNIX Programmer's Guide* for information about the event filtering API.

**Filtering Events for Display in the Main Workspace**

You can select which events are displayed in the main workspace or the Event History workspace by activating filters.

To activate a filter, follow these steps:

1. Click **Filter Control** in the **Options** pull-down menu in the Event Display or Event History application’s window.

   The Filter Control dialog box is displayed.

![Filter Control Dialog Box](image)

   **Figure 27. Filter Control Dialog Box**

The Filter Control dialog box contains two sections. On the left is the **Available Filters in File** section, which lists each filter in the selected file and provides a short description. On the right is the **Active Filters List**, which displays the name, description, and full path name of all filters that are currently active.
2. If you want to select a different input file, click **File List** and select another filter directory or file from the File Selection dialog box, shown in Figure 28. To see the file in the specified directory, click **Filter** in the File Selection dialog box. After you have made your selection, click **OK** to close this dialog box.

3. To activate a filter, select a filter from the Available Filters in File Filter/Description list in the Filter Control dialog box and click **Activate**. The filter information is copied to the Active Filters List, and the filter is activated immediately.

4. If you want display the filtered events from the `/usr/OV/log/ovevent.log` file for display by the Event History application, click **Display Events** in the **Query** pull-down menu in the main workspace window. See “Viewing the Event Log” on page 121 for more information.

   If there are too many objects selected for inclusion in the filter, the filter will not be activated, and the following message will be displayed:

   *Sieve creation failed. Error in Object creation.*
If filters that retrieve logged events have more than 40 host names defined as one particular object, for example, IP_ADDRESS, you will receive the following error message:

Error receiving logged event.

To resolve the error, edit the filter to reduce the number of objects (IP_ADDRESS).

When activating more than one simple filter, the logical operator OR is used. If you want the criteria of both filters to be used, use the Compound Filter Editor to combine them and then activate the compound filter.

See “Creating Compound Filters” on page 146 for more information.

To display a selected filter from the Available Filters in File Filter/Description list, click Display/Edit below the list of available filters on the Filter Control dialog box. The Filter Editor dialog box is displayed. From this dialog box, you can make changes to the following filter criteria:

- The filter name
- The filter description
- The enterprise name
- The generic and specific event numbers
- The object from which the trap is to be sent
- The time range, including a starting date and time and a stopping date and time
- The threshold values, including frequency and time interval

Note, however, that you cannot edit a filter from the Available Filters in File list if that filter has been placed in the Active Filters List. If you need to make changes to an active filter, first deactivate the filter, then click Display/Edit to access the Filter Editor.

See “Using the Filter Editor” on page 143 for more information about editing filters.

All filters you activate in a workspace can be saved in a configuration file in the $USER/$HOME directory. See “Saving the Event Workspace Configuration” on page 119 for information about saving workspace configuration.

**Deactivating an Event Filter**

To deactivate an event filter, follow these steps:

1. Click Filter Control in the Options pull-down menu in the Event Display or Event History application’s window.
   
   The Filter Control dialog box is displayed.

2. Select the filter name from the Active Filters List and click Deactivate.

   The filter is immediately deactivated, and the number of events being displayed might be affected.

**Activating a Filter in a Dynamic Workspace**

To activate a filter for a dynamic workspace, click Dynamic Workspace in the Create pull-down menu in the main workspace.

Refer to the Tivoli NetView for UNIX User’s Guide for Beginners for more information about creating a dynamic workspace.
Activating a Trap-to-Alert Filter (AIX Only)

If you are using the Tivoli NetView host connection, you can specify that selected events are to be converted to alerts and forwarded to the Tivoli NetView for z/OS product. Trap-to-alert filters permit selected events to pass to the tralertd daemon, which converts events and traps to alerts. You can create and activate filters that limit the number of traps to be processed by the tralertd daemon. All trap-to-alert filters are stored in the /usr/OV/conf/tralertd_default.filter file.

Note: You must have root authority to activate trap-to-alert filters because these filters affect the configuration of the tralertd daemon.

To activate a trap-to-alert filter, follow these steps:

1. In the Options pull-down menu, click the Event Configuration —> Trap to Alert Filter Control: SNMP.

The Trap-to-Alert Filter Control dialog box is displayed as shown in Figure 29.

2. If the file name in the Input File field at the top of the Trap to Alert Filter Control dialog box is not the one you want to use, change it by using one of the following methods:
   - Type a new file name in the Input File field
   - Click the File List button and select a directory or file name from those displayed in the File Selection dialog box. Select the OK button to close this dialog box.

3. From the Available Filters in File list, select the name of the filter you want to activate and click the Activate button.
The filter file name is copied to the Active Filters List and the filter is immediately activated.

If there are too many objects selected for inclusion in the filter, the filter will not be activated, and the following message will be displayed:
Sieve creation failed. Error in Object creation.

**Displaying a Trap-to-Alert Filter (AIX Only)**
Suppose you select a filter from the Available Filters in File list, but before you activate it you want to ensure that it is the one you want to use. Select the name of the filter and click **Display/Edit**. Depending on whether the selected filter is simple or compound, the appropriate Filter Editor dialog box is displayed. The filter criteria are displayed in the fields of this dialog box. If you need to change a field, make the changes here, then click either **Save As** button to save the changes under a different filter name or **OK** to close the filter editor.

**Deactivating a Trap-to-Alert Filter (AIX Only)**
To deactivate a trap-to-alert filter, select its name in the Active Filters List, then click **Deactivate**. The Event Display application will no longer filter events based on this filter. Deactivating a filter might affect the number of events being displayed.

**Creating Cron Table Entries for Trap-to-Alert Filters (AIX Only)**
If you want certain filters to be activated and deactivated on a regular and frequent basis, the buttons in the top section of the Trap to Alert Filter Control dialog box. However, you might spend much of your time keeping track of which filters to activate and deactivate, and when.

To automate filter activity, use the bottom section of the dialog box, Cron Table Filter Control. This operation enables you to create and modify cron table entries. The cron daemon uses these entries to control the tralertd daemon’s use of selected filters. Using this operation gives you more precise control of the alerts forwarded to the host program.

**Activating and Deactivating Cron Table Entries:** In the Cron Table Filter Control section of the Trap to Alert Filter Control dialog box, you can enter activation and deactivation parameters for filters that appear in the Active Filters List in the top half of the dialog box. For example, to activate a filter called **EveryWednesday** on Wednesdays at 7 a.m. and deactivate it at 3 p.m. on Wednesdays, follow these steps:

1. Ensure that the filter is in the Active Filters List. If it is not there, select it from the Available Filters in File list and click **Activate**.
2. Select the filter in the Active Filters List.
3. In the Cron Table Filter Control section, enter 07:00 in the Activation field and click check for Wednesday. Then, enter 15:00 in the Deactivation field, and select the check button for Wednesday. The 24-hour clock is used for entries in these fields.
4. Click **Add to Cron**. The activation and deactivation entries for the selected filter appear in the Trap-to-Alert Cron Table Entries field.
5. Click **Close** to close the Trap to Alert Filter Control dialog box.

**Modifying a Cron Table Entry:** You can modify a cron table entry by selecting it in the Trap-to-Alert Cron Table Entries field, specifying new activation or deactivation parameters, and then clicking **Modify**.
**Sorting Cron Table Entries:** To sort the entries in the Trap-to-Alert Cron Table Entries field, click **Sort**. Entries can be sorted in any of the following ways:
- By activation day of the week
- By deactivation day of the week
- By activation hour
- By deactivation hour
- By filter name (the default)

To remove an entry from the list, click Trap-to-Alert Cron Table Entries field, then click **Remove**.

**Using the selectfilter Command (AIX Only)**

You can use the `selectfilter` command in a terminal window to accomplish the same tasks you can perform by clicking **Options —> Event Configuration —> Trap to Alert Filter Control: SNMP** in the Tivoli NetView main menu. These tasks include the following:
- Activate a filter immediately.
- Deactivate a filter immediately.
- Add an entry to the cron table that will issue the `selectfilter` command at regular intervals to activate or deactivate filters.
- Remove a selectfilter entry from the cron table.

The following example activates a filter called `newFilter` every Wednesday at 5:30 p.m.

```
selectfilter -f /usr/OV/filters
  -r newFilter
  -s /usr/OV/sockets/tralertd.socket
  -a 1 -t 17:30 3
```

For more information about the parameters of the `selectfilter` command, refer to the man page.

---

**Configuring Events**

Every enterprise has a Management Information Base (MIB) that describes operations that can be performed on that enterprise's devices in the network. Enterprises can specify traps that they expect to receive from agents that support their MIBs. You can configure the events supplied with the MIB to provide additional, more specific information about the status of network objects.

**Advantages to Configuring Events**

Event configuration offers the following advantages:
- You can format a trap to display information that is meaningful to you.
- You can format a trap that is forwarded to the Tivoli Enterprise Console to display meaningful information.
- You can associate an action with a trap by specifying the commands that are to be executed when the management station receives an event. Configuring actions enables you to automate some fault management procedures and to restrict the amount of event information to be displayed.
- You can associate severities with an event. These severity levels are displayed in the upper-left corner of event cards. You can search on events by severity level.
- You can create new event notification categories for filtering events.
• You can create new, additional actions to specify further processing that the operator should manually perform when an event is received.
• You can provide a message window to be displayed when selected events are received. This option is available when you add, copy, or modify an event. The maximum number of windows you can display is ten.
• You can access the trap-to-alert filter control to convert events to alerts that are sent to the Tivoli NetView for z/OS product (AIX only).

**Customizing Traps**

To use Event Configuration —> Trap Customization: SNMP in the Options pull-down menu, the following conditions must be true:

- You must be logged in as a root user.
- The node for which you want to configure an event must support SNMP.
- The enterprise-specific MIB for which you want to configure events must be loaded into the Loaded MIBs database.
- You must understand the definition of and purpose for the MIBs on which you want to configure events. Review the documentation provided by the MIB vendor about the enterprise-specific events included with their product.

**Note:** Trap customization can cause security problems. See “Avoiding Security Problems Caused By Trap Customization” on page 156 for more information.

When an event is configured, it is added to the /usr/OV/conf/C/trapd.conf file. When this event is received from an agent, the information in the /usr/OV/conf/C/trapd.conf file is used to format the trap information that is logged in the /usr/OV/log/trapd.log file.

In addition, the Event Display application reads the event information from the /usr/OV/conf/C/trapd.conf file and formats it for display in the event cards or list. The Event Display application updates symbol status based on status events. In addition to defining the event as a status event, the object must have object or symbol status source.

By default, the Event Display application cannot update objects that have compound status source. You can determine or change the symbol’s status source by clicking Edit —> Modify/Describe Symbol in the object context menu. To enable the Event Display application to update a symbol’s status if the status source is compound, change the value for the overrideCompoundStatus resource in the /usr/OV/app-defaults/Nvevents file to TRUE. When the value for the overrideCompoundStatus resource is TRUE, the Event Display application updates all symbols for the object without having to manually change the status source for each symbol.

See “Indicating Symbol Status” on page 45 for more information.

To delete configured events, select the event and click Delete. Another way to delete events is to edit the trapd.conf file and delete, or comment out, selected configured events.

**Steps**

To configure an event that provides you with a more readable message, follow these steps:

1. In the Options pull-down menu, click Event Configuration —> Trap Customization: SNMP.
The Event Configuration dialog box is displayed, as shown in Figure 30 on page 155. Click Help for information about the fields and buttons in this dialog box.

2. If desired, use the Configure Event Categories dialog box to define event categories. To display this dialog box, select the Configure Categories button.

3. In the Event Identification section, select an enterprise by selecting an item in the selection list. If the enterprise whose event you want to configure is not in this list, click Add New Enterprise and enter the enterprise name and object ID in the Add New Enterprise dialog box. Click Add to add the new enterprise and close the dialog box.

To view a list of the Tivoli NetView product’s internally generated events, enter the /usr/OV/bin/event -l command at the command line.

4. Select the event that you want to configure from the selection list in the Event Identification section. If the event already exists, click Modify to display the Modify Event dialog box. Otherwise, you can create a new event by clicking Add or Copy. Clicking Add displays the Add Event dialog box. Clicking Copy displays the Copy Event dialog box.

5. Complete the fields in the Add Event, Copy Event, or Modify Event dialog box. Click Help in each dialog box for detailed information on completing the fields. Click OK button to return to the Event Configuration dialog box.

6. Repeat steps 3 through 5 until all events have been configured.

7. Use the Configure Additional Actions for Operator dialog box to define additional actions that the operator should manually perform when an event is received. Click Configure Additional Actions to display the dialog box.
8. Click **OK** or **Apply** in the Event Configuration dialog box to apply any changes you have made. Click **Cancel** if you want to undo the changes.

**Note:** Behaviors such as list sizes and default values can be modified in the `/usr/OV/app-defaults/XNm` file.

**Avoiding Security Problems Caused By Trap Customization:** The ovactiond, nvcorrd and actionsvr daemons filter out all non-alphanumeric characters except for the minus sign (-) and the decimal point (.) to prevent a security problem. All characters not falling into this set are replaced with an underscore (_). The following characters are modified: $ ` & @ # % ^ < > \ = [ ] - " . If a minus sign or decimal point is encountered, it is escaped (for example, preceded by a backslash (\)) as a precaution. If any non-alphanumeric character is encountered, and filtering is not disabled, a message is entered into the appropriate log file (/usr/OV/log/nvcorrd.alog, /usr/OV/log/ovactiond.log, and /usr/OV/log/nvaction.alog).

You can customize this behavior by using an environment variable called `AdditionalLegalTrapCharacters`. If you set this variable to `disable`, then no filtering is done. If you set this environment variable to a string containing non-alphanumeric characters, then the filtering will allow those characters to also pass through the filter, but they are escaped. The best method for setting an environment variable for an ovspmd controlled daemon is to put the definition of the environment variable into the `/usr/OV/bin/netnmrc.pre` file, then stop and restart all the daemons (using netnmrc to restart).

**Note:** The contents of the environment variable `AdditionalLegalTrapCharacters` must be quoted, and some of the characters must be escaped to avoid being operated on by the shell or the system. The following assignment demonstrates a valid string for the `AdditionalLegalTrapCharacters` variable:

```plaintext
AdditionalLegalTrapCharacters=
! @ # / $ ^ & * ( ) _ + \ / = [ ] - " ; , . > < #: \n
Verifying Trap Customization

If you want to verify the change, look at the `/usr/OV/conf/C/trapd.conf` file. The next time this event is received from an agent, the log message will contain the message you specified.

If you want to ensure that the message will be displayed as you specified, you can issue the `snmptrap` command at the command line. The `snmptrap` command issues an SNMP trap based on the parameters you specify in the command. For example, to send the Tivoli NetView enterprise-specific trap number 59160427 to a host named `host1` from an agent named `agent1`, enter the following command:

```
snmptrap host1 "" agent1 6 59160427 ""
```

For more information about the `snmptrap` command, refer to the man page.

Another way to see the event you just created is to use the `event` command. This command sends an event to the trapd daemon. Specify the specific number of the event as follows:

```
event -E 59160428
```

For more information about the `event` command, refer to the man page.
Using the addtrap Command to Configure Events

You can type the \texttt{addtrap} command at the command line to configure an event without using the Options \textrightarrow Event Configuration \rightarrow Trap Customization: SNMP menu item. The \texttt{addtrap} command creates a trap and adds the new trap to the /usr/OV/conf/C/trapd.conf file.

You can also use the \texttt{mib2trap} command to retrieve special comments from a MIB definition file and create a shell script that contains a series of \texttt{addtrap} commands. Then, you can execute the shell script to add the new traps to the /usr/OV/conf/C/trapd.conf file.

For more information about the \texttt{mib2trap} command, refer to the man page.

If there is no enterprise definition for the trap, the new enterprise definition is added. If a trap exists with identical enterprise-object-ID, generic-trap, and specific-trap values, the \texttt{addtrap} command updates the existing trap with the new information.

**Example**

The following example illustrates the \texttt{addtrap} command that adds a trap for the IBM 6611 Router to the /usr/OV/conf/C/trapd.conf file.

\begin{verbatim}
addtrap -n ibm6611
  -i 1.3.6.1.4.1.2.6.2
  -g 6 -s 16 -o A -t 3
  -c "Status Events"
  -f !
  -F '\$E \$G \$S \$T'
  -S 4
  -C xecho
  -A 'Status Event received from 6611 agent $E$G $S'
  -e nodeDown
  -E msg
  -V 'PRINTF ("Node \%s down", \$V2)'
\end{verbatim}

This command specifies the following information:

- **\texttt{-n}** The enterprise name is \texttt{ibm6611}.
- **\texttt{-i}** The enterprise ID is \texttt{1.3.6.1.4.1.2.6.2}
- **\texttt{-g}** The generic trap number is \texttt{6}.
- **\texttt{-s}** The specific trap number is \texttt{16}.
- **\texttt{-o}** The trap is sent from an agent, in this case, the 6611 router agent.
- **\texttt{-t}** The object that generates the trap is to be assigned a status of Critical on the map.
- **\texttt{-c}** This is a status event.
- **\texttt{-f}** A specified action (see \texttt{-C} and \texttt{-A} below) will be performed by the management system when this trap is received.
- **\texttt{-F}** The enterprise name (\$E), generic (\$G) and specific (\$S) event numbers, and the time-stamp (\$T) are displayed in the event cards or list.
- **\texttt{-S}** The trap is a Severity 4 (Critical) trap.
- **\texttt{-C}** The \texttt{xecho} command is activated when this event is received.
- **\texttt{-A}** The following arguments are passed to the Tivoli NetView product with this event:
- Event text ('Threshold Event received from 6611 agent')
- The enterprise name ($E)
- The generic trap number ($G)
- The specific trap number ($S)

-e This event is forwarded to the Tivoli Enterprise Console with an event class of nodeDown.

-E This event is forwarded to the Tivoli Enterprise Console containing the event text specified by the -V flag.

-V Trap variable 2, the host name, is substituted in the event text for the %s format specifier when this event is forwarded to the Tivoli Enterprise Console.

For more information about the addtrap command, refer to the man page.

---

**Displaying a Warning Window for Events**

To display a pop-up window, specify a shell script when a specific event occurs that executes the ovxbeep or ovxecho command when a specific event occurs. If you execute the ovxbeep command in the shell script, an error dialog box is displayed with an audible alarm. If you execute the ovxecho command in the shell script, an error dialog box is displayed without an audible alarm. The shell script must export the display to the appropriate workstations before executing the ovxecho or ovxbeep commands, and the xhost command must have been run on the workstations where the pop-up window is to be displayed.

You specify the name of the shell script in the Optional Command and Argument format section of the Event Configuration dialog box.

For example, say you want to display a pop-up window when NodeA or NodeB fail. For NodeA you want to include an alarm. You also want to send an electronic-mail notice of the failure. Here are the steps:

1. In the Options pull-down menu, click Event Configuration —> Trap Customization: SNMP.
2. .
3. On the Event Configuration dialog box, select or enter the following:
   - Enterprise name: netview6000
   - Event: Specific 58916865
4. Click Modify. The Modify Event dialog box is displayed.
5. Enter the following in the Command for Automatic Action field:
   - `<ShellScriptPath> $2`
6. Click OK to close the Modify Event dialog box.
7. Click OK to close the Event Configuration dialog box.

**Note:** If you filter an event for which you have configured a command for automatic action, the actions specified in the shell script will still be executed. If the shell script executes the ovxbeep or ovxecho command, for example, an error dialog box is displayed even though the event has been filtered.

**Example Shell Script**

Following is the shell script used in the previous example.
#!/bin/ksh
# example.sh
#
# Shell script for node down trap from the netview6000 enterprise
# (specific = 58916865). Displays warning messages and sends e-mail.

export DISPLAY=NodeA.austin.tivoli.com:0
export DISPLAY=NodeB.austin.tivoli.com:0

if [ $1 = NodeA.austin.tivoli.com ]; then
  /usr/OV/bin/ovxbeep $1" is down"
  echo $1" is down" | mail oper1@manager.austin.tivoli.com
fi

if [ $1 = NodeB.austin.tivoli.com ]; then
  /usr/OV/bin/ovxecho $1" is down"
  echo $1" is down" | mail oper2@manager.austin.tivoli.com
fi

The $2 passes to the script the name of the device that generated the alert. The shell script checks the $2 flag to see whether it is NodeA or NodeB that generated the alert. If it is NodeA, the shell script calls a program, /usr/OV/bin/ovxbeep that displays a window and an audible alarm. If it is NodeB, the shell script calls the program, /usr/OV/bin/ovxecho that displays a window without a sound. For either node, an electronic-mail notice is sent to the addresses specified in the shell script.

See the /usr/OV/prg_samples/nnm_examples/beeper/beep_951x sample shell script for more examples.

---

**Converting Events to Alerts (AIX Only)**

If you are using the [Tivoli NetView host connection](#), you can edit the events, or traps, that are converted to SNA alerts and forwarded to the host program.

Use the [Alert Editor](#) to define the SNA alert for a trap that must be forwarded to the host program. You can start the Alert Editor from the **Options —> Event Configuration —> Trap Customization: SNMP** menu item, which displays the Event Configuration dialog box. The Alert Editor button is on the right side of this dialog box.

Use the Alert Editor to perform the following tasks:
- Configure the trap-to-alert mapping for selected events
- Define the alert for each expected trap by specifying the following information:
  - Type of event
  - Description
  - Probable or failure causes
  - Qualifiers
  - Recommended actions to take
- Delete an alert
- Write alerts to the tralertd.conf file, which can be edited
- Check errors to make sure that the defined alert is valid

The Alert Editor also enables you to modify existing information on alerts through text fields and buttons. When the events are converted and sent to the host program, pertinent alert information is displayed on Tivoli NetView for z/OS screens.
Using the addalert command (AIX Only)

You can use the `addalert` command to add an alert definition to the tralertd.conf file without using the Tivoli NetView graphical interface. Enter the `addalert` command and its parameters at the command line to perform the same function as selecting Options —> Event Configuration and clicking Alert Editor.

The following example of the `addalert` command adds an alert definition for a Bay Networks agent.

```
addalert -o .1.3.6.1.4.1.10
   -l bayalert
   -g 6
   -s 0
   -t 1
   -d 1400
   -p 0202
   -q 13
   -m "$1"
   -f 0202
   -a 1320
```

This command specifies the following information:

- `-o` The enterprise ID is .1.3.6.1.4.1.10
- `-l` The alert label is bayalert.
- `-g` The generic trap number is 6.
- `-s` The specific trap number is 0.
- `-t` The alert type is Permanent (1).
- `-d` The Generic Alert subvector (X'92') contains a description for generic alert 1400, Loss of Electrical Power.
- `-p` The Probable Causes Alert subvector (X'93') contains probable cause 0202, Internal Power Control Unit.
- `-q` The Detailed Data Alert subvector (X'98') contains the detailed data entry Status Code.
- `-m` The message passed to the host program resides in $1.
- `-f` The Failure Caused Alert subvector (X'96') contains failure cause 0202, Internal Power Control Unit.
- `-a` The Recommended Actions subvector (X'81') contains recommended action 1320, Check Cable Connection and Retry.

For more information about the `addalert` command, refer to the man page.

Refer to Tivoli NetView for UNIX Host Connection for instructions on using the Alert Editor.

Sending Alerts to the Host Program (AIX Only)

When a trap is received by the trap-to-alert conversion process and it matches an active filter, the trap is converted to an SNA alert and forwarded to the Tivoli NetView for z/OS product.

The goal is to send all information about a trap to the host program. However, sometimes the trap contains too much information to be sent in one piece. In that
In this case, the trap is saved in the tralertd database and assigned a Log ID, which is sent to the host program in the alert. Tivoli NetView for z/OS uses the Log ID to issue the `gettrap` command within a RUNCMD to request complete trap information for the incomplete alert.

The trap information sent to the host program differs based on whether the trap is IBM enterprise-specific, non-IBM enterprise-specific, or generic. Refer to *Tivoli NetView for UNIX Host Connection* for more information about sending alerts to Tivoli NetView for z/OS.
Chapter 6. Managing Network Configuration

One of the challenges of network management is keeping track of all the devices on a network and ensuring that you have current information about how they are configured. Current configuration information can help you perform the following tasks:

- Make sure that all devices are configured correctly.
- Resolve network connectivity, performance, and service problems.
- Customize polling intervals to regulate network traffic and collect necessary information.
- Configure SNMP proxies to manage non-SNMP devices.

This chapter describes the following configuration management tasks:

- “Discovering the Network”
- “Monitoring the Network Using Polling” on page 169
- “Monitoring Network Configuration” on page 174
- “Retrieving MIB Configuration Information” on page 180
- “Enabling and Disabling Polling” on page 169
- “Configuring SNMP Nodes” on page 181
- “Configuring a Backup Manager” on page 184

Discovering the Network

The Tivoli NetView product provides two ways to discover IP networks and enables the discovery of open topology networks and Cisco routers. Network discovery provides you with a database of network configuration information. This section describes the following:

- “Automatic Network Discovery”
- “Discovering Open Topology Networks” on page 166
- “Discovering Cisco Routers” on page 167
- “Configuring Symbol Creation Time and Buffer” on page 167
- “Increasing the ovwdb Cache Size” on page 168

Automatic Network Discovery

The Tivoli NetView product uses an automatic network discovery process to generate and maintain a network topology database. The more nodes on the network that support an SNMP agent, the more efficient this discovery process will be, and the more complete and accurate the resulting configuration information will be. Discovery starts with the management station, then proceeds to discover everything up to the first set of routers. Subnets beyond that are unmanaged. Use a seed file to include additional devices in the initial discovery process.

Information Retrieved

When a new node is discovered, it is added to the topology database and also to the list of nodes that is being monitored. If the newly discovered node supports an SNMP agent, information about its system configuration is retrieved and stored in the database. Table 14 on page 164 shows the information that is retrieved:
Table 14. Configuration Information Retrieved From Nodes During Discovery

<table>
<thead>
<tr>
<th>Information</th>
<th>MIB Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System description</td>
<td>sysDescr</td>
<td>Includes the full name and version of the system’s hardware type, software operating system, and networking software.</td>
</tr>
<tr>
<td>System object ID</td>
<td>sysObjectID</td>
<td>Identifies the network object’s place in the MIB hierarchy.</td>
</tr>
<tr>
<td>Forwarding status</td>
<td>ipForwarding</td>
<td>Indicates whether this entity is acting as an IP gateway to forward datagrams received by, but not addressed to, this entity.</td>
</tr>
<tr>
<td>IP address table</td>
<td>ipAddrTable</td>
<td>Lists addressing information relevant to this entity’s IP addresses.</td>
</tr>
<tr>
<td>Interface table</td>
<td>ifTable, ifNumber, ifSpeed</td>
<td>Lists interface entries by number.</td>
</tr>
<tr>
<td>System location</td>
<td>sysLocation</td>
<td>Indicates the physical location of this network object.</td>
</tr>
<tr>
<td>System contact</td>
<td>sysContact</td>
<td>Lists the person to contact for this network object and tells how to contact that person.</td>
</tr>
</tbody>
</table>

Once the node has been discovered, these MIB values are polled periodically. Any changes are reflected in the topology database.

If the Tivoli NetView product is configured to work with a relational database, you can store IP topology data in a relational database and use the relational database tools to create reports.

Refer to the Tivoli NetView for UNIX Database Guide for more information about transferring IP topology data to a relational database.

You can store additional, enterprise-specific information with each node and network. This information, used with the predefined data generated by the Tivoli NetView product, can give you a clearer picture of your network’s configuration.

Note: The Tivoli NetView product sometimes thinks that devices with multiple interfaces are routers and displays the router symbol.

Turning Off Automatic Discovery
In the Options pull-down menu, click Topology/Status Polling Intervals: IP to turn automatic discovery off. Click Discover New Nodes to turn polling off.

Discovering Cisco Routers
The Tivoli NetView product takes advantage of the Cisco CDP Cache MIB (if available) on routers for next hop information. This greatly improves discovery in a Cisco environment and provides a significantly better chance at discovering the unnumbered serial links promptly. When a CDP MIB is available, netmon will not read the IP Routing table on routers, which can take up to 4 or 5 minutes on a busy router.
What netmon does with the CDP Cache MIB information:

- The default discovery mode for netmon is Local. That is, netmon discovers and manages the local subnet and displays the next hop network as Unmanaged. If the CDP Cache MIB information is available, netmon automatically attempts to discover the other side of a serial link without having to rely on a seed file entry specifying it. The other end of the serial link is discovered as Unmanaged, and the serial link is not displayed until both ends are managed.

- You can then extend the discovery as usual by managing the networks and routers. This will initiate a New Node discovery poll, and after a few minutes, you will see the serial links added in. To expedite this behavior, you can issue demand poll requests to the routers.

- Isolated islands will exist when the next hop routers cannot be queried using SNMP.

- For non-Cisco routers, NetView attempts to discover the next hop for unnumbered serial links using other means.

Discovering z/OS™ Systems

The Tivoli NetView program discovers z/OS systems that are running the IBM z/OS V1R4 Communications Server TCP/IP stack and its SNMP TCP/IP Subagent.

The Tivoli NetView program identifies the individual nodes on the submap using a z/OS system symbol. The Tivoli NetView program queries the z/OS Communications Server SNMP TCP/IP Enterprise-specific MIB as part of the discovery and daily configuration polls to define the OVwDB fields for the node.

The oid_to_type file defines the Z flag that identifies z/OS systems.

See Appendix A, “Tivoli NetView Internal Traps”, on page 281 for information about the traps that are generated for these MVS™ fields if they change.

OVwDB fields for z/OS systems: The Tivoli NetView program provides the following OVwDB fields for z/OS systems:

<table>
<thead>
<tr>
<th>OVwDB field</th>
<th>MIB Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVS SystemName</td>
<td>ibmMvsSystemName</td>
<td>The name of the MVS image on which the TCP/IP Subagent is active.</td>
</tr>
<tr>
<td>MVS SysplexName</td>
<td>ibmMvsSysplexName</td>
<td>The name of the sysplex in which the MVS image is running.</td>
</tr>
<tr>
<td>TcpipProcName</td>
<td>ibmMvsTcpipProcname</td>
<td>The MVS procedure name (Procname) of this TCP/IP instance</td>
</tr>
</tbody>
</table>

Use one of the following methods to work with the OVwDB fields:

- The ovobjprint or ovobjdump commands
- The object properties menu
- The Web console submap explorer
- OVwDB application programming interface (API) calls

Using a Seed File to Control Network Discovery

When the Tivoli NetView product is started for the first time, the default management region is the management station on which the Tivoli NetView
product is operating and any networks to which it is attached. The discovery process generates the topology map by working outward from the management station up to the first set of routers.

You can define a management region by using a seed file. A seed file contains a list of host names or IP addresses of SNMP nodes within your administrative domain. Using a seed file forces or restricts the discovery process to generate the topology map beginning from nodes other than the management station.

See the Tivoli NetView Release Notes, Version 7.1 for more information.

**Discovering Open Topology Networks**

The Tivoli NetView product uses specially created applications to manage networks that use protocols other than IP. These applications pass topology information to the Tivoli NetView product in the form of enterprise-specific SNMP traps or API calls.

An open-topology discovery application can be started whenever the Tivoli NetView product is started, or it can be started only when a node that supports a protocol other than IP is discovered on the network. The information acquired by the open-topology discovery application is stored in the general topology database on the manager, and can be displayed on a submap along with information about IP network nodes.

**Discovering Topology Using the Openmon Application**

You can use the Tivoli NetView openmon application to discover and load topology information from an openmon agent into the Tivoli NetView product. In the **Administer** pull-down menu, click **Openmon Application** to start the openmon configuration program to create an application that interacts with an agent that represents a specific topology. The application stores the information obtained from the agent in the GTM database. The topology is displayed along with other network topologies, such as IP, on the Tivoli NetView graphical interface. The topology information is correlated with IP topology information, wherever possible, to determine whether an IP object can also be identified as having an association with the topology stored in the GTM database. In the **Administer** pull-down menu, click **Openmon Application** to start, query, or stop the application.

When the openmon functions receives additional attribute information for the topology objects, openmon stores the attribute information along with the objects in the OVW object database. To view the attribute information click **Display Object Information** in the **Tools** pull-down menu or use the **ovobjprint** command.

Refer to the **openmon** man page for additional information.

The Novell NetWare topology is an example of a topology that openmon creates. The openmon application interfaces with one or more Novell ManageWise Export Services to discover and load the Novell Netware topology into the GTM database.

The NetWare topology is represented on the root map with a Novell icon. When you double-click on the Novell icon, a separate icon is displayed for each ManageWise Export Service. Each ManageWise Export Service contains one object for the IPX network view and one object for the servers view. Here is a description of the NetWare topology views:
IPX network
Contains all the IPX network segments connected by the routers that ManageWise discovered.

Servers
Contains the icons for all the NetWare servers that ManageWise discovered.

Router
Contains the icons for the router adapters and the running software functions, such as IP, IPX routing functions, file servers, and so forth.

Segment
Contains all the NetWare servers, requesters, hubs, and routers in the segment.

Server
Contains the icons for the adapter and all the running functions such as file server, printer server, and so forth.

Requester
Contains the icon for the adapter.

Hub
Contains the icons for the adapter and the running functions such as the hub function, file server, and so forth.

Discovering Cisco Routers
The Tivoli NetView product takes advantage of the Cisco CDP Cache MIB (if available) on routers for next hop information. This greatly improves discovery in a Cisco environment and provides a significantly better chance of discovering the unnumbered serial links promptly. When a CDP MIB is available, netmon will not read the IP Routing table on routers, which can take up to 4 or 5 minutes on a busy router.

What netmon does with the CDP Cache MIB information:

- The default discovery mode for netmon is Local. That is, netmon discovers and manages the local subnet and displays the next hop network as Unmanaged. If the CDP Cache MIB information is available, netmon automatically attempts to discover the other side of a serial link without having to rely on a seed file entry specifying it. The other end of the serial link is discovered as Unmanaged, and the serial link is not displayed until both ends are managed.

- You can then extend the discovery as usual by managing the networks and routers. This will initiate a New Node discovery poll, and after a few minutes, you will see the serial links added in. To expedite this behavior, you can issue demand poll requests to the routers.

- Isolated islands will exist when the next hop routers cannot be queried using SNMP.

- For non-Cisco routers, the Tivoli NetView product attempts to discover the next hop for unnumbered serial links using other means.

Configuring Symbol Creation Time and Buffer
The ipmap application draws the IP topology maps that represent your network in the graphical interface. When a node is discovered, ipmap stores it in a buffer. When the buffer reaches its threshold, ipmap draws the symbols for the nodes in the buffer onto the map. In other words, it dumps the buffer to the map. Depending on the size of your network and the speed at which nodes are discovered, you can improve the performance of ipmap by altering the size of the buffer and how often it is dumped.
Using **Describe Map** from the **File** pull-down menu, you can configure the number of symbols to be created at one time and how often they are to be created. When a node is discovered, it remains in the buffer until the maximum number of nodes arrives or the set time has expired. Normally, the batch size is never reached before the time expires.

The default for the buffer size is 150 nodes. The maximum value allowed is 1000 nodes and the minimum is 1. The synchronization buffer uses the same value set for the buffer size (when no timer is present).

The default and recommended setting for the timer is 3 seconds. More than 3 seconds results in poor response time and less time results in slower overall performance of ipmap. The maximum value allowed is 3600 seconds and the minimum is zero (0).

The size of the buffer should vary in proportion to the speed of the processor.

**Steps**
To change the buffer size or timer, follow these steps:
1. In the **File** pull-down menu, click **Describe Map**.
2. In the Configurable Applications selection list, click **IP Map**.
3. Click **Configure For This Map**.
4. Enter the number of symbols in the field *How many symbols should be created at one time?*. 
5. Enter the time in the field *How Often (in seconds) should IP Map create symbols?*.
6. Click **Verify** to check what you entered.
7. Click **OK** to apply the change and close the dialog box.

**Increasing the owpdb Cache Size**

The owpdb daemon acts as a caching daemon for the object information stored in the object database. You can control the number of objects maintained in the cache. Increasing the cache size will improve your CPU performance, particularly for networks containing more than 5000 objects.

**Changing the Size**

Change the size of the owpdb’s cache from the default of 5000 objects to a number larger than the number of objects in the owpdb database. To get the number of objects in the owpdb database, use the following command:

```
ovobjprint | head -1
```

To change the size of the owpdb’s cache, follow these steps:
1. Exit all Tivoli NetView windows. All opened sessions of the ovw application are closed during execution of the last step.
2. Enter `serversetup` on the command line. The IBM Tivoli NetView Setup: Menu window is displayed.
3. Click **Configure —> Set options for daemons —> Set options for topology, discovery, and database daemons —> Set options for owpdb daemon**. The Set options for owpdb dialog box is displayed.
4. Complete the dialog box fields as follows:
   * Enter a number in the Number of objects to hold in cache field. Enter a number larger than the number of objects defined in the owpdb database. That enables the cache to grow to the maximum size if needed.
Click Yes or No in the Use port to receive requests over tcp? field. Click Help on the dialog box for additional information.

Monitor the size of the database and adjust the cache size as necessary. If RAM size and paging space are not a problem, using a cache size of zero allows the cache size to grow to an unlimited size and shrink as needed.

If the Tivoli NetView product discovers a network that exceeds the available paging space, the operating system might stop a process, including owdb, to relieve paging space. See “Monitoring File System and Paging Space” on page 209 for information about monitoring your paging space.

### Monitoring the Network Using Polling

The Tivoli NetView product has comprehensive polling capabilities that enable you to view the up-to-date status of your network. The Tivoli NetView product can use ping (ICMP echo) requests to poll devices for status, or SNMP GET requests to determine the status of all interfaces on a device.

Many aspects of the Tivoli NetView polling configuration can be customized to fit your network environment. This section describes the following:

- “Enabling and Disabling Polling”
- “Using SNMP for Status Polling” on page 171
- “Understanding Router Fault Isolation” on page 172

### Enabling and Disabling Polling

In the Options pull-down menu, click **Topology/Status Polling Intervals: IP** to turn polling on or off for all IP nodes in the management region. When you click the **Options —> Topology/Status Polling Intervals: IP** option, the Topology/Status Polling Configuration window is displayed as shown in **Figure 31**.

#### Enabling Topology Polling and Discovery

The Topology/Status Polling Configuration window provides check buttons that enable you to turn polling and discovery on or off. The **Enable Polling and Discovery Settings** check button must be selected for the settings of the other buttons in the dialog box to take effect and for the values in the SNMP Configuration Dialog Box to take effect.

Click the **Enable Polling and Discovery Settings** check button to turn polling and discovery on or off according to the following variations:
Turn all polling intervals and discovery on by first clicking *Enable Polling and Discovery Settings*, then clicking *Poll for Status*, *Discover New Nodes*, and *Poll for Configuration Changes* (all three) in the dialog box. All polling intervals and discovery are turned on.

Individually, turn polling intervals and discovery on by first clicking *Enable Polling and Discovery Settings*, then clicking *Poll for Status*, *Discover New Nodes*, or *Poll for Configuration Changes* by clicking the associated check button. For example, if you click the Enable Polling and Discovery Settings check button then set Poll for Status on, Discover New Nodes off, and Poll for Configuration Changes on, status polling and configuration checks occur, but new node discovery does not occur.

Turn Poll for Status off, Discover New Nodes off, and Poll for Configuration Changes off by deselecting the *Enable Polling and Discovery Settings* check button. The individual settings of the other check buttons in the dialog box do not take effect if the Enable Polling and Discovery check button is turned off. If this check button is not selected, the values set in the SNMP Configuration Dialog Box do not take effect.

### Polling for Status, New Node Discovery, and Configuration Changes

After clicking *Enable Polling and Discovery Settings* as described in the previous section, the other check buttons enable you to poll for status, discover new nodes, and poll for configuration changes.

**Note:** The Discover New Services button functions independently of the *Enable Polling and Discovery Settings*. See “Discover New Services” on page 171 for more information.

You can individually turn on the following:

- **Poll for Status**
  When you poll for status, nodes are polled for status states. If they do not respond to status polling during the amount of time specified in the Node Down Delete Interval field in the SNMP Configuration Dialog box, they are automatically deleted from the object database. The default is 7 days. To change the time period specified in the Node Down Delete Interval field, the *Poll for Status* button must be selected.

- **Discover New Nodes**
  When you discover new nodes, the Tivoli NetView product polls existing SNMP nodes to determine if new nodes exist. The frequency of this polling is determined by whether you use an auto-adjusting polling interval or specify a fixed polling interval in the [SNMP Configuration Dialog box](#). Use an auto-adjusting polling interval to send less polling traffic to the manager once most of the network is discovered. Auto-adjusting polling is the default.

- **Poll for Configuration Changes**
  SNMP nodes can be polled at specified time intervals to check their configuration status. You can regulate this polling by clicking *Poll for Configuration Changes* and specifying a value in the Fixed Polling Interval field in the [SNMP Configuration Dialog box](#)

By default, the Tivoli NetView product checks the configuration once per day. It checks the /usr/OV/conf/ovsnmp.conf database to determine how frequently to poll the node for status. This file contains default configuration values that you can change; see “Configuring SNMP Nodes” on page 181 for more information.

Configuration checking provides the following information for selected network nodes:
- Change in contact or location
- Forwarding IP packets change
- Interface added
- Interface deleted
- Incorrect routing by a node
- Link address change
- Mismatch of link address
- Network mask change
- Node name change
- Object identifier change
- Undetermined link address

If you want to set or change polling intervals for individual nodes, click either SNMP Configuration in the Options pull-down menu or Data Collection & Thresholds: SNMP in the Tools pull-down menu.

Discover New Services
The service monitor application is implemented as the servmon daemon. The servmon daemon discovers and monitors services on Tivoli NetView managed nodes. Use the Discover New Services button to enable or disable the servmon daemon. Note that when this button is selected or cleared, trap CSP_EV is issued, which eliminates the need to stop and restart the servmon daemon.

Using SNMP for Status Polling
Agents in the network are regularly polled by NetView to update the status of the objects in the topology map. Normally, the Tivoli NetView product sends ping requests to all interfaces on a device to determine their status. Certain network devices have interfaces whose status cannot be determined from a ping. These include devices with unnumbered interfaces, such as ATM devices.

For these devices, the Tivoli NetView product can be configured to determine the status using SNMP queries for the ifAdminStatus and ifOperStatus values of the interfaces. This behavior is set in one of three ways:

1. Automatically, if the node has at least one unnumbered interface.
2. Explicitly, by setting the IP address or range of addresses in your netmon seed file.

   Use Server Setup application (Configure —> Set options for daemons —> Set options for topology, discovery, and database daemons —> Set options for netmon daemon) by creating an SNMP Status entry using the Network Monitor Seed File Editor. You can also edit the netmon.seed file using a text editor; the prefix $ denotes an SNMP Status entry. The same rules apply for IP address ranges and pattern matching characters (wildcards) as those for negative seed entries.

3. Explicitly, using the P switch in the /usr/OV/conf/oid_to_type file against classes of devices

   If the Tivoli NetView product is configured to use SNMP to poll for status, the demand poll will report the values of the ifAdminStatus and ifOperStatus for each interface of the node. If you still want to use a ping request to poll for status on these nodes during a demand poll, specify the -E switch in the

   /usr/OV/lrf/netmon.lrf file.


Refer to the Tivoli NetView Release Notes, Version 7.1 for information on the format
of the `oid_to_type` file. Refer to the *Tivoli NetView for UNIX Administrator’s Reference*, the *Tivoli NetView Release Notes*, Version 7.1, or the man pages for more information on the netmon daemon.

**Understanding Router Fault Isolation**

The Router Fault Isolation function reduces the diagnostic cost of locating a network problem, especially when the network becomes partitioned as a result of the failure. Router Fault Isolation simplifies the notification action by issuing only one summary alert identifying the router nearest the fault.

When you use the Tivoli NetView program to manage a network with a high proportion of nodes to routers, Router Fault Isolation can significantly reduce the number of Node Down events that are false alarms. Router Fault Isolation detects which nodes are actually down and which nodes are simply unreachable because the router fault is occluding them from the management station. Router Fault Isolation relies on connectivity tests and responds instantly to dynamic routing changes.

During a network failure, the Tivoli NetView management workstation cannot reach devices or interfaces in the failed portion of the network. For example, a failed router interface can make other portions of the network, such as subnets, invisible or inaccessible. That portion of the network is blocked and considered unreachable.

Without Router Fault Isolation, the Tivoli NetView program creates numerous events during a network failure and again after the failure is corrected. This event proliferation:

- Increases the difficulty of determining the original cause of the network failure
- Slows network traffic considerably with the large number of status polls to the occluded area
- Creates performance problems and unreliable status reports if the events are forwarded to the Tivoli Enterprise Console

When a node or interface is down, Router Fault Isolation first checks the status and accessibility of the router interfaces connected to that subnet. During the router check, each interface and its subnet are analyzed. An interface that does not respond triggers checks of the interface and any connecting routers. Router Fault Isolation generates an event for each Router Down or Router Marginal event. The Tivoli NetView maps display unreachable networks and router nodes or interfaces as white symbols. Note that non-router nodes and interfaces in unreachable subnets are not changed to Unreachable (white).

When active, the Router Fault Isolation feature generates the following events to alert users to important status changes:

*Table 16. Events generated by Router Fault Isolation.*

<table>
<thead>
<tr>
<th>Events</th>
<th>Network Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router Marginal</td>
<td>At least one router interface is down. At least one other interface on that router is up.</td>
</tr>
<tr>
<td>Router Down</td>
<td>All interfaces are not responding, but at least one connected subnet is reachable. (The router is not in an occluded region.)</td>
</tr>
<tr>
<td>Router Unreachable</td>
<td>The network management workstation cannot query the router because it is an occluded region.</td>
</tr>
</tbody>
</table>
Table 16. Events generated by Router Fault Isolation. (continued)

<table>
<thead>
<tr>
<th>Events</th>
<th>Network Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router Up</td>
<td>All the interfaces have responded successfully. This event is issued on initial discovery and following a recovery from one or more interfaces being down.</td>
</tr>
<tr>
<td>Network Unreachable</td>
<td>All router interfaces in the subnet have stopped responding.</td>
</tr>
<tr>
<td>Network Reachable</td>
<td>After one interface is successfully polled, the network is once again reachable.</td>
</tr>
</tbody>
</table>

Router Fault Isolation also suppresses polls and status events for all non-router nodes and interfaces in unreachable subnets. After a partition is repaired, the first successful status poll from inside an unreachable subnet triggers a recovery. To speed the initiation of recovery, you can also manually ping any node in an unreachable region.

To reduce the false signals from a Node Down event for a device in an area with Unreachable status, the Tivoli NetView program does not generate Node Down or Interface Down events for any node in the area with Unreachable status. The first Interface Down event that triggers an evaluation that results in declaring that the status of the subnet is Unreachable is also suppressed. Status polling to end nodes in subnets with Unreachable status is suppressed by default.

If the Tivoli NetView program is not managing any routers in a particular subnet, then the Tivoli NetView program can determine when that subnet is unreachable. It does this using a probabilistic algorithm, which determines when it is highly likely that the subnet is unreachable. The Tivoli NetView program automatically uses this algorithm for subnets where there are no managed routers. However, this algorithm only determines the reachability of the subnet. If it is unreachable, Node Down and Interface Down events, including the first event, are not generated.

Router Fault Isolation and Mid-Level Manager can function together and object status should remain accurate. However, in some cases both could poll the same nodes. This causes extra processing because routers are polled by both, and in some situations, this causes extra network traffic.

RFI Configuration

To configure the RFI mode, use the Server Setup application. Select Configure —> Set options for daemons —> Set options for topology, discovery and database daemons —> Set options for netmon daemon and set the Router Fault Isolation Mode.

There are three modes for RFI that can be configured:

- **Disabled**
  - No attempt is made to determine reachability or root cause. Routers generate node status events, instead of the root cause router status events.

- **Router Fault Isolation Mode**
  - By dynamically evaluating the status of routers, The Tivoli NetView program determines the reachability of subnets and the root cause of the partition or problem.

- **Probabilistic Mode**
  - By dynamically evaluating the status of members of a subnet, the Tivoli NetView program determines whether it is more likely that the
The subnet itself is unreachable or whether the devices are down. This mode is disabled if the subnet contains less than a configured number of managed devices. This mode is automatically used for subnets with no managed routers if RFI Mode is active. You can fine-tune this algorithm using properties defined in the netmon.conf configuration file. Refer to the /usr/OV/conf/netmon.conf file for more information.

The Server Setup application also provides the ability to treat ambiguous router interfaces that are not responding in unmanaged subnets as though their status is either Unreachable or Down. In the Set options for netmon daemon window of Server Setup, set the Router Fault Isolation: Treat ambiguous nonresponding router interfaces in unmanaged subnets as field to either Unreachable or Down. See the /usr/OV/doc/RouterFaultIsolation.htm file for information about using this option.

**Stopping the Router Fault Isolation feature**
The Router Fault Isolation feature is active by default in the Tivoli NetView program. To turn this feature off, use the -K 0 option for netmon. To control the suppression of polling traffic to routers (including unreachable routers), use the -k option for netmon. Refer to the netmon man page for more information. For a detailed explanation of Router Fault Isolation, see the description in the /usr/OV/doc/RouterFaultIsolation.htm file.

**Monitoring Network Configuration**
There are many reasons to keep track of the configuration of objects on your network. The following list provides some configuration information you might need:

- What the IP and non-IP addresses are for a node or nodes
- What types of interfaces a node supports, and the status of each
- How a node is connected to the network
- Whether two nodes show the same address for a third node
- Whether a particular service has been installed on a remote node

The following sections describe how you can obtain this information.

**Listing IP Addresses for Remote SNMP Nodes**
To list IP and non-IP addresses associated with a remote SNMP node, click a node on a submap, then click the Network Configuration —> Addresses in the Monitor pull-down menu. This operation collects information that you might otherwise have to obtain by looking at numerous configuration files. The following information is provided about each interface this object has with the network:

- The index, or MIB instance, of the interface on the selected node. You must have this value if you plan to set up MIB data collection.
- The name of the interface
- The IP address of the interface
- The network mask
- The network address
- The link address, if any
Checking Configured Interfaces

In the Monitor pull-down menu, click Network Configuration —> Interfaces to list information about interfaces on remote SNMP nodes. This information can help you resolve performance problems because it provides statistics on incoming and outgoing SNMP node traffic and associated errors. It can also help you resolve connectivity problems, because it lists the status of interfaces.

Clicking Network Configuration —> Interfaces in the Monitor pull-down menu provides the following information:

- The index, or MIB instance, of the interface on the selected node. You must have this value if you plan to set up MIB data collection.
- The name of the interface.
- The type of interface, for example, loopback, Ethernet, FDDI.
- The maximum transmission unit (MTU) size. This is the largest packet size that can be sent unfragmented.
- The status of the interface, which can be up, down, or testing (no operation packets can be passed through the interface).
- The total number of input packets and the number of erroneous input packets received.
- The total number of output packets and the number of erroneous output packets sent.

For hubs and bridges, each entry in the table corresponds to a port on the hub or bridge.

Viewing Routing Table Information

To obtain routing table information for selected remote SNMP nodes, use the Network Configuration —> Routing Table in the Monitor pull-down menu. This operation can help you resolve connectivity problems. The following information is provided by this operation:

- Destinations. The default destination is a route used by the system when it cannot find a specific route.
- The name of the next gateway between the selected node and the destination.
- The type of connection, as follows:
  - Direct—to a directly connected local area network (LAN)
  - Remote—Through a remote gateway
  - Other
- The name of the interface that is used to reach the destination.

You can also list information about gateway routing tables by entering the following command in at the command line:

```
rnetstat -r <host name>
```

Refer to the man page for more information about the `rnetstat` command.

Obtaining ARP Cache Information

The Address Resolution Protocol (ARP) cache is helpful in resolving connectivity problems, because it can tell you whether two nodes have a different link address than a third node. To obtain this information for selected remote SNMP nodes, click Network Configuration —> ARP Cache in the Monitor pull-down menu. The following information is displayed:
The name or IP address of the destination node
The link address associated with the destination node
The interface name of the selected node that is used to access the destination node

You can list the ARP Cache table for a selected SNMP node by entering the following command at the command line:

```
netstat -A <host name>
```

For more information about the `netstat` command, refer to the man page.

### Listing Configured Protocols

You can determine the protocols that a node is configured to support. For example, if a user is having a problem with a remote Simple Network Management Protocol (SNMP) node, click Monitor —> Network Configuration —> Services to determine the following information:

- The communication transport protocol, which can be either Transmission Control Protocol (TCP) or User Datagram Protocol (UDP).
- The port to which the protocol is bound.
- The protocol for which the node is listening (for example, SNMP, telnet, or Network File System (NFS)). If this field is blank, the service is unknown.

To list the configured protocols for a remote SNMP node, enter the following command:

```
netstat -S <host name>
```

For more information about the `netstat` command, refer to the man page.

### Discovering Services on Nodes

Part of managing a network involves knowing which nodes provide important services, such as DB2 servers and WebSphere Application servers, and monitoring the status of these services. The servmon daemon provides the ability to discover and monitor these services. The service monitor must be configured to specify which services to discover and monitor, and how to perform these tasks. See the `/usr/OV/conf/servmon.conf` file to determine which services are discovered and monitored by default.

The service monitor is implemented as the servmon daemon. The servmon daemon is started by default. See “Enabling and Disabling the servmon Daemon” on page 177 for information about controlling the servmon daemon.

Servmon responds to asynchronous events that it receives from the trapd and nvcold daemons. See “Scheduling Service Discovery” on page 178 for more information about scheduling the service monitor.

Service SmartSets are created by the service monitor containing nodes that have a particular service. For each service that is discovered on a node by the service monitor, nodes become members of service SmartSets if servmon discovers a service on the node.

For each service that is discovered by the service monitor, servmon creates a Boolean field in the node object as designated by the Node Field field of the...
configuration file entry. The field value is set to TRUE, and the node automatically becomes a member of the designated service SmartSet.

If a service is removed from the node, servmon removes the designated Boolean field from the node after it has determined that the service has been unavailable for the interval specified by the Delete Services Down For interval in the nmpolling window.

Service objects are represented at the interface level of the node. The label assigned to a service object is the Service Label Name field from the corresponding servmon configuration file entry. The status of each service is checked at the interval specified by the corresponding servmon configuration file entry to update the service objects.

Service objects that are in the Critical state for the configured maximum allowed down time are automatically deleted from the object database. The Boolean field of the corresponding node object is also deleted, effectively removing the node from the service SmartSet.

### Working With the Servmon Daemon

This section provides the following information:

- “Enabling and Disabling the servmon Daemon”
- “Scheduling Service Discovery” on page 178
- “Understanding the Service Monitor Configuration File” on page 178
- “Using Custom Plug-in Modules for Discovery and Status Checking” on page 180

### Enabling and Disabling the servmon Daemon

The servmon daemon is enabled by default when the Tivoli NetView product is started. Use the following procedures to disable or enable the servmon daemon.

#### Disabling the servmon Daemon:

1. Enter serversetup on the command line. The IBM Tivoli NetView Setup: Menu window is displayed.
2. Click Configure —> Set options for daemons —> Set options for topology, discovery, and database daemons —> Set options for servmon daemon. The IBM Tivoli NetView Setup: Options window is displayed.
3. Clear the Enable servmon daemon check box.
4. Click OK.

#### Enabling the servmon Daemon:

1. Enter serversetup on the command line. The IBM Tivoli NetView Setup: Menu window is displayed.
2. Click Configure —> Set options for daemons —> Set options for topology, discovery, and database daemons —> Set options for servmon daemon. The IBM Tivoli NetView Setup: Options window is displayed.
3. Select the Enable servmon daemon check box.
4. Click OK.
Specifying How Long a Down Service is Monitored
The service monitor automatically deletes objects from the object database that
have been in a Critical state for a specified amount of time.

By default, the service monitor stops monitoring the status of a service after it has
been down for 7 days. Use the following procedure to change this value:
1. Enter `xnmnmpconf` on the command line. The SNMP Configuration window is
displayed. Select the Default entry.
2. Type a value for Service Down Delete Interval. Valid values are 1 - 31, and d
   (days) must be specified after the number, separated by a space.
3. Click Replace.
4. Click OK to close the SNMP Configuration window.

Scheduling Service Discovery
Use the following procedure to specify how often the service monitor should poll
the network nodes to determine if a new service has been added:
1. Enter `xnmnmpconf` on the command line. The SNMP Configuration window is
displayed. Select the Default entry.
2. Specify one of the following values for Service Discovery Interval:

<table>
<thead>
<tr>
<th>Valid Value</th>
<th>Time Qualifier</th>
<th>Example Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 — 44640</td>
<td>m (minutes)</td>
<td>120 m</td>
</tr>
<tr>
<td>1 — 744</td>
<td>h (hours)</td>
<td>12 h</td>
</tr>
<tr>
<td>1 — 31</td>
<td>d (days)</td>
<td>5 d</td>
</tr>
<tr>
<td>1 — 4</td>
<td>w (weeks)</td>
<td>2 w</td>
</tr>
</tbody>
</table>
3. Click OK to close the SNMP Configuration window.

Understanding the Service Monitor Configuration File
The default configuration file contains entries for common services, and these
services are declared available if the basic port test succeeds. You can specify
custom modules for service discovery and status checking. See “Using Custom
Plug-in Modules for Discovery and Status Checking” on page 180 for more
information.

The service monitor configuration file specifies the following information:
• The services that you want to discover
• The ports to test for a specified service (required for entries that do not specify a
custom plug-in discovery or status test module)
• The service SmartSets you want to create
• The name that you want to use to label the service objects that are created
• Optional custom plug-in discovery and status test modules
• How often you want to check the status of a service

The default servmon configuration file is `/usr/OV/conf/servmon.conf`, and it
contains entries for some common services that nodes might have. You can modify
this file or create new configuration files. If you want to use a configuration file
other than the default file, specify the full path of the configuration file using the `-c
<config_file_name>` switch.
Syntax and format errors are logged in the `/usr/OV/log/servmon.log` file, and the entry in which the error occurred is ignored.

**Configuration File Entry Statements:** The format for each configuration file entry is:

```
[Node Field][TCP Port][Service SmartSet][Service Label Name]
[Discover Test][Status Test][Discover Node Selection Criteria].
```

where:

**Parameter**

**Description**

**Node Field**

The name of the Boolean field to create on a node object if the service is successfully discovered. A value must be specified for this parameter. The value cannot contain a space.

**TCP Port**

Either a comma-separated list of the TCP port numbers to test on each node, or arguments for a custom plug-in, launchable test if one is specified. A value must be specified for this field unless a custom test is specified for both the Discover test and Status test fields.

**Service SmartSet**

The name of the service SmartSet to create if a service is discovered. If a value is not specified, a service SmartSet is not created. The value cannot contain a space.

This field is used as the node selection criteria for monitoring the status of the service. If a value is not specified for this field, the status of the service defined by this entry is not checked.

**Service Label Name**

The symbol label name assigned to the service object created as a result of the service being discovered. A value must be specified unless the Service SmartSet field is not specified. The value cannot contain a space.

**Discover Test**

The name of the optional, custom plug-in module to use to discover a service. Specify the full path of the launchable application. The same launchable application can be specified as the one that is specified in the Status Test field. The use of Java™ JAR file custom extensions is reserved for use only by the Tivoli NetView product.

**Status Test**

The name of the optional, custom plug-in module to be used to check the status of a service that has already been discovered. Specify the full path of the launchable application. The same launchable application can be specified as the one that is specified in the Discover Test field. The use of Java JAR file custom extensions is reserved for use only by the Tivoli NetView product.

**Discover Node Selection Criteria**

Specifies the nodes to test during service discovery. Specify either the name of any existing SmartSet, or an asterisk (*) to specify all managed IP nodes. The value cannot contain a space.

**Status Interval**

Specifies how often (in minutes) to check the status of this service on nodes.
known to have this service. The maximum value that can be specified is 44640
minutes. A value of 0 specifies that no status checking should be performed for
this service.

Using Custom Plug-in Modules for Discovery and Status
Checking
You can specify a custom plug-in module for the Discover Test and Status Test
fields in a service monitor configuration file entry. The same plug-in module can
be used for both tests. Custom plug-in tests can consist of any thread-safe,
re-entrant launchable script or executable file.

Retrieving MIB Configuration Information
Certain MIB variables store information that provides a summary of the
configuration of a selected network node or nodes. The operations described in
this section enable you to conveniently retrieve the values of frequently accessed
MIB variables.

Displaying MIB Interface Information
To display the current values of a network object’s interface MIB variables, select a
symbol on the submap, then click MIB Values —> Interface Info in the Monitor
pull-down menu. The following information is displayed:

- **ifDescr**
  Information about the manufacturer, product, and version of the hardware
  interface

- **ifType**
  Type of interface

- **ifMtu**
  Size of largest datagram that can be sent or received over this interface,
  specified in octets

- **ifSpeed**
  Estimated current bandwidth of the interface, in bits per second (bps)

- **ifPhysAddress**
  Address of protocol layer at the protocol layer immediately below the
  network layer in the protocol stack, in octets.

- **ifAdminStatus**
  Desired state of the interface (up, down, or testing)

- **ifOperStatus**
  Current operational state of the interface

Displaying MIB System Information
To display the current values of a network object’s system MIB variables, select a
symbol on the submap, then click MIB Values —> System Information in the Monitor
pull-down menu. The following information is displayed:

- **sysDescr**
  Information about the name and version of the hardware, software
  operating system, and networking software

- **sysObjectID**
  The node that is being managed

- **sysUpTime**
  Time, in hundredths of a second, since the network management portion of
  a system was last initialized


sysContact
Contact person for this managed node, and how to contact that person

sysName
Administrative name (fully qualified domain name) assigned to this node

sysLocation
Physical location of the node

---

Configuring SNMP Nodes

The Tivoli NetView default configuration values are stored in the /usr/OV/conf/ovsnmp.conf_db file. You can make the following changes to this file by clicking the SNMP Configuration in the Options pull-down menu:

- Change the default SNMP configuration.
- Change the netmon daemon’s status polling intervals.
- Configure different community names for different nodes in your network. The community name used by the management application must match the name that is configured on the agent of the managed node.

**Note:** Be careful when configuring a Get community name for a SmartSet. Remember that the netmon daemon needs a valid community name to discover nodes, but a node cannot be in a SmartSet until it is discovered. For discovery purposes, the community name should be set in the Specific Nodes or IP Address Wildcards section on the SNMP Configuration dialog box if it is not the value specified in the default configuration.

- Configure SNMP proxies to manage non-SNMP devices.
- Configure a specific node or a group of nodes to have different values than the default configuration.
- Control polling for a specific node or a group of nodes according to the following variations:
  - To turn new node discovery on or off, click Discovery Poll.
  - To turn auto-adjusting polling on or off, click Auto Adjust.
  - Control the configuration check polling interval.
- Discover new nodes as managed or unmanaged by clicking Discover Node(s) Managed.

In a distributed network environment, make sure that you configure managed network devices to enable SNMP communication from client machines as well as from the manager workstation. If you do not properly configure the network device, all SNMP requests from the clients, such as MIB applications, MIB browsers, and so forth, will time out with no response from the device.

**Note:** The polling configuration values are set to the default settings when automatic map generation is restarted or the databases are cleared.

---

How an Application Uses SNMP Configuration

When an application initiates an SNMP request, the SNMP APIs look for a configuration entry in the node list to query the node. If the application cannot find an entry, it looks for the first IP address wildcard entry in the network list that matches the IP address of the node. If an entry exists, then it is used. If a network list entry does not exist for the node, the application uses the default SNMP configuration.
In a distributed network environment, SNMP requests issued from the client are directly transmitted on the network from the client. That is, the requests do not go through the server first. Therefore, clients must have their community names configured to enable SNMP requests on a node.

When an SNMP request is made for a node name, the node name must resolve to an IP address. You must have the node entry in the /etc/hosts file or the domain name server. When the node name does not resolve to an IP address, the community name public is used to send the SNMP request.

**Description of the SNMP Configuration Dialog Box**

Figure 32 shows the SNMP Configuration dialog box.

![SNMP Configuration Dialog Box](image)

The SNMP Configuration dialog box contains two main sections. The top section contains three selection lists that display current SNMP configuration values for:
- Specific nodes
- A group of nodes
- SmartSets
- The default SNMP configuration for your network

The bottom section contains text entry fields in which you can specify new values for entries in the top section.
Note: If you change the value for configuration checking to a shorter interval, you must perform a demand poll on a node to make that change effective. Otherwise, you would need to wait 24 hours or to update the rate of the daily configuration check.

When you select an item from any of the three selection lists in the top section, the current parameters for that entry are displayed in the bottom section. You can then view, delete, or modify the parameters.

The order of the three selection lists, and the order of entries within each list, illustrates the precedence order in which entries are searched when a node is queried.

Note: The parameters displayed in the selection lists can be tailored by the X11 resource xnmsnmpconf.summaryList. If this resource is set to True, only the Target, Community, Set Community and Proxy parameters are displayed in the selection lists. Otherwise, all parameters are displayed in the selection lists.

Steps

The values you enter for individual nodes override the values set in the Global Default section. The Enable Polling and Discovery Settings button must be selected in the Topology/Status Polling Configuration dialog box for the values you enter in the SNMP Configuration dialog box to apply. To change the configuration for an SNMP node, follow these steps:

1. In the Options pull-down menu, click SNMP Configuration.
2. Select the item you want to change from one of the selection lists. The configuration for the item you selected is displayed in the SNMP Parameters section.
3. Change the following configuration values as appropriate.
   - Proxy
   - Target
   - Community
   - Set Community
   - Timeout
   - Retry count
   - Remote Port
   - Status Polling
   - Node Down Delete Interval
   - Fixed Polling Interval
   - Configuration Polling Interval
   - Number of Route Entries
   - Service Discovery Interval
   - Service Down Delete Interval
   - Discovery Poll
   - Auto Adjust
   - Discover Node(s) Managed

   Click Help for information about these fields.
4. Click Replace to display the new values in the appropriate selection list.
5. Click OK or Apply to apply the changes.

To configure a specific node, a group of nodes, or a SmartSet to have different configuration values than the default configuration, follow these steps:
1. In the Options pull-down menu, click **SNMP Configuration**.

2. Enter the values in the text entry fields in the SNMP Parameters section. If you specify a specific node, you only need to fill in the fields that have different configuration values than the default configuration.

3. Click **Add** to add the new values to the appropriate clickion list. If the target is a regular host name or an IP address, the configuration is added to the Specific Nodes clickion list. If the target is an IP address with wildcards, the configuration is added to the IP Address Wildcards clickion list. If the target is a SmartSet, the configuration is added to the SmartSets clickion list.

4. Click either **Apply** or **OK** to apply the changes.

If you cannot set the isSNMPProxied general attribute in the map database, add the target to the map. The target must exist in the map database and have a selection name.

---

**Configuring a Backup Manager**

With Tivoli NetView, you can segment a large network and create individualized spheres of control for several management stations. Objects outside an operator’s sphere-of-control are unmanaged by that operator’s management station. When an object is unmanaged, Tivoli NetView no longer polls that object for status and configuration changes. You can have numerous Tivoli NetView programs on the network. Each one can be configured so that there is little duplication of management network traffic.

Using the **Backup Configuration dialog box** you can configure objects as manager nodes (Tivoli NetView products) or as managed containers, and then specify which manager is managing each container. A **managed container** is a collection of objects and is designated as being managed by one or more managers or not managed at all.

The entire network can be separated into various containers using the concept of a **partitioned Internet map**. Each manager can be designated to manage a subset of the containers. Managed containers that are defined and managed by a remote Tivoli NetView managing system are unmanaged.

Each manager checks on the **status** of the other known managers and notifies the backup manager when a manager is disabled or when the manager is restored. The operator can then **manage the backup session**.

See “Managing a Backup Session” on page 190 for more detailed information.

Depending on the configuration of the local manager, it can manage all or part of the disabled manager’s containers.

A submap labeled ManagerSubmap is displayed on the root submap. The Manager submap contains the symbol for each manager node that is discovered on the network. The Manager submap’s only function is to group all managers together in one place, allowing you to quickly **determine current backup sessions** for the selected remote manager.

See “Determining Current Backup Sessions” on page 191 for more detailed information.
Configuring Manager-Container Associations

To configure nodes as managers and specify the containers they manage, do one of the following:

- Create a seed file that specifies which associations to make
- Use the Backup Configuration dialog box

There is no manager-to-manager communication. Therefore, to configure manager-container associations across multiple managers, create a definitive seed file and distribute it to all managers. This ensures the same configuration file is used for all managers. If you change a map or a manager-container association on a manager, change all managers. The manager-container configurations are global only on the local system. Use the Backup Configuration dialog box to display and modify the associations already configured.

Manager-Container Characteristics

Managers and containers must have the following characteristics to be valid:

- The name specified must be the selection name of an object in the object database.
- The specified manager name is a node (the isNode capability must be set to True).
- The specified container name must be a Location, Internet, Network, or Segment object.

Using a Seed File to Configure Managers

You can create a seed file to pass to the backup process. A seed file is used by the Tivoli NetView product to determine which nodes are managers, as well as how manager-container associations should be made. It defines the relationship between managers, containers, and backup managers. The seed file is passed to the backup process by modifying the command line for the backup process in the application registration file, /usr/OV/registration/C/backup. Find the line:

```
Command -Shared -Initial -Restart "${BackupDir:-/usr/OV/bin}/backup";
```

Add the seed file name to the end of the line as follows:

```
backup [-s /path/seed_file_name]";
```

Save the seed file in the /usr/OV/conf directory. Include the full path name. Invoke the seed file through the registration file and not from the command line.

Seed File Format:

The seed file consists of three fields per line:

```
active manager  container  backup manager
```

Where:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>active manager</td>
<td>Is the name of a manager. This is the active manager for the container object.</td>
</tr>
<tr>
<td>container</td>
<td>Is the container object name.</td>
</tr>
<tr>
<td>backup manager</td>
<td>Is the name of the backup manager.</td>
</tr>
</tbody>
</table>

You can designate more than one backup manager as shown in the following example. Names must be enclosed in double quotation marks:

```
"bos.tivoli.com" "192.10.30" "rudd.tivoli.com" "cowan.tivoli.com"
```
Where:

<table>
<thead>
<tr>
<th>bos.tivoli.com</th>
<th>Is the manager node.</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.10.30</td>
<td>Is the container node.</td>
</tr>
<tr>
<td>rudd.tivoli.com</td>
<td>Are backup manager nodes.</td>
</tr>
<tr>
<td>cowan.tivoli.com</td>
<td></td>
</tr>
</tbody>
</table>

If bos should go down, both backup managers will start managing the specified container objects.

**Seed File Process:** The seed file is processed line-by-line. The manager name and the container name are read. These objects are validated to make sure that they actually exist and that they have the appropriate characteristics. If the objects are valid, the manager object’s capability field, isManager, is set to True. Associations between the manager and container are activated. The container object is added to the manager’s selected containers list and the manager object is added to the container’s selected managers list. The container is checked to see if it should appear as managed or unmanaged by the ipmap application. After those associations are made, the associations between the backup managers and the containers are made.

If the seed file has been updated and changed or the necessary managers and containers were not discovered when the seed file was initially read, click Backup —> Read Seedfile in the Administer pull-down menu to reread the seed file. This option adds associations to those that already exist. Associations are not deleted.

**Seed File Example**

In the following example, M1, M2, and M3 are manager nodes in a network. Each manager has the following backup seed file configured:

```
"M1" "Backbone" "M2"
"M1" "North_America_Network" "M2"
"M1" "Latin_America_Network" "M3"
"M1" "South_America_Network" "M3"
"M1" "Antarctica_Network"
"M2" "Africa_Network" "M1" "M3"
"M2" "Eastern_Europe_Network" "M1"
"M2" "Western_Europe_Network" "M1"
"M3" "Asia_Network" "M1" "M2"
"M3" "Australia_Network" "M1"
"M3" "Middle_East_Network" "M1" "M2"
```

When M1 is started, Backbone, North_America_Network, Latin_America_Network, South_America_Network, and Antarctica_Network are automatically managed. Africa_Network, Eastern_Europe_Network, Western_Europe_Network, Asia_Network, Australia_Network, and Middle_East_Network are automatically unmanaged.

When M2 is started, Africa_Network, Eastern_Europe_Network, and Western_Europe_Network are automatically managed. All the other containers are unmanaged.

When M3 is started, Asia_Network, Australia_Network, and the Middle_East_Network are automatically managed. All the other containers are unmanaged.
In this example, each manager has a subset of the whole network to manage. M1 is acting as the backup for all containers of M2 and M3. M2 and M3 are backing up specific containers. The Antarctica_Network is not backed up at all.

**Using the Backup Configuration Dialog Box**

Use the Backup Configuration dialog box, shown in Figure 33, to designate nodes as managers and add, display, and modify the configuration of backup managers and containers. Define and configure remote manager nodes on the local manager node so they can be backed up.

The Backup Configuration dialog box displays a list of all configured manager nodes (nodes that have the isManager capability field set to True) and a list of all the container nodes known by the Tivoli NetView product.

You must have a read-write map to configure backup managers. Only one Backup Configuration dialog box can be opened for each object database.

**Designating Nodes as Managers:** To designate nodes as managers, follow these steps:

1. In the Administer pull-down menu, click Backup —> Backup Configuration. The Backup Configuration dialog box is displayed.

![Backup Configuration Dialog Box](image-url)
2. Enter the name of the node in the Selection field or select an object on the submap and click **Select From Map**. When you select an object from the map, the object characteristics are checked. If it is a valid manager, its selection name is placed in the manager selection box.

3. Click **OK**. The Manager Configuration window is displayed.

4. Click **True** next to the isManager field. The manager node name is added to the list of managers in the Managers field. When using a seed file, the isManager field is automatically set to **True**.

5. Click **OK** or **Apply**.

6. Repeat the steps for each node you want to designate as a manager.

Once you have designated nodes to be managers, you can make your manager-container associations.

Modifications to the isManager capability field causes the Tivoli NetView product to reevaluate the managed or unmanaged state of all containers in the Selected Containers list.

**Adding Manager-Container Associations:** Configure on the local managing system the remote managers and which containers they are actively managing. Also add the containers you want the local manager to backup if the active manager for those containers should go down. The manager-container associations that you configure apply to the current system’s database only.

You can make a manager-container association even though the selected node does not have the isManager capability set to true.

Dividing the backup management of containers between managers prevents a sphere of control, which could be an entire network, from being placed on one manager.

**Steps:** To configure manager-container associations, follow these steps:

1. In the **Administer** pull-down menu, click **Backup —> Backup Configuration**. The Backup Configuration dialog box is displayed.

2. Do one of the following:
   - Select a Manager or a Container from either list.
   - Enter the name of the node in the Selection field.
   - Select an object on the submap and click on the **Select From Map** button. When you select an object from the submap, the object characteristics are checked. If it is a valid manager or container, its selection name is placed in the appropriate selection box.

3. Click **OK** if you select an item from the list.

   **If a manager was selected:**
   a. The Manager Configuration window is displayed.
   b. Do one of the following:
      - Select a container from the Available Containers list. This list displays all of the available containers that are not associated to the selected manager. Click **Select** to create an association between the manager and the selected container. The container name is removed from the Available list and added to the Selected Containers list.
• Type a container object name in the field below the Selected Container list and click **Add to List**. The container name is added to the Selected Containers list.

• Select an object from the map, click **Select from Map**, then click **Add to List**. The container name is added to the Selected Containers list.

The Selected Containers list displays all of the containers associated with the selected manager. Selecting containers from the Selected Containers list can affect the managed or unmanaged state of that container on the manager.

c. Select the type of association to be made for the selected manager and container.

Click **Active** if you are configuring associations between a manager and the containers that the manager is managing. There can be more than one manager configured as active for each container.

Click **Backup** if you are configuring associations between a manager and the containers that this manager is going to manage if the active manager should go down. Multiple backup managers for a single container and multiple backup containers per backup manager are permitted.

d. Repeat the steps for each manager-container association.

If a container was selected:

a. The Container Configuration window is displayed.

b. Do one of the following:

• Select a manager from the Available Managers list. This list displays all of the available managers that are not associated with the selected container. Click **Select** to create an association between the container and the selected manager. The manager name is removed from the Available list and added to the Selected Managers list, which displays all of the managers associated with the selected container.

• Type a manager name in the field below the Selected Managers list and click **Add to List**. The manager name is added to the Selected Managers list.

• Select an object from the map, click **Select from Map** and then **Add to List**. The manager name is added to the Selected Managers list.

c. Select the type of association to be made (active or backup).

d. Repeat the steps for each manager-container association you want to add.

4. Click **Apply** to activate your associations.

5. Click **OK** to close the Container Configuration window.

**Deleting Associations:** Associations can be deleted by selecting a container from the Selected list and clicking **Remove**. Deleting containers from the selected list can affect the managed or unmanaged state of that container on the local manager.

To delete managers from the Managers list, select a manager and click **False** next to the isManager capability field.

If you remove associations that contain non-manager nodes, the manager is moved from the Selected Managers to the Available Managers list when you select Remove. If you exit the Container Configuration window and then return, the node is no longer on the Available Managers list because the capability was not set to isManager.
Determining Remote Manager Status
The status of remote managers is determined by Node Up and Node Down traps sent by netmon to the local manager.

When the local manager receives a Node Down trap for a remote manager, all active containers for that remote manager are examined. If a container is also part of the local manager’s sphere of control, no further action is taken because the local operator already has control of that container. By default, the local manager receives notification of a disabled manager. A Manager Down message box is displayed, and the operator can initiate actions from the message box to manage the backup session.

When the local manager receives a Node Up trap for a remote manager, a Manager Restored message box is displayed stating the remote manager’s name and the fact that it is now active. The operator can initiate actions from the message dialog box to manage the backup session. No further action is taken by the backup process until the operator returns the container to unmanaged state by closing the container submap from the Navigation Tree window, the Manager Submap, or through the Manager Restored message box. At that time the managed or unmanaged state of the container is reevaluated.

Note: Unmanaging remote managers or submaps and containers in which the remote manager objects reside prevents the local manager from being notified when the remote manager is disabled, because the local manager no longer polls those objects.

Manage-Unmanage Rules
When the configuration changes for a container, for example, a container is removed from the list of Selected Containers associated with a manager, the Tivoli NetView product determines whether to automatically manage or unmanage the container.

When the configuration changes for any manager, for example, the isManager capability changes from true to false, the managed or unmanaged condition for all of that node’s managed containers is updated. Here is how the Tivoli NetView product handles status changes:
- If the local manager and the given container are associated, the container is managed.
- If the container is not associated with any manager, the container is managed by the local manager.
- If an association exists between this container and any remote node where the isManager capability is set to True, the container is unmanaged by the local manager.
- If all remote associations are with nodes where isManager is set to False, the container is managed by the local manager.

Managing a Backup Session
This section describes the actions you can take when you receive notification of a remote manager status change. This section also describes how you can customize the backup default actions, and how you can determine what containers a backup manager is managing.

Responding To a Manager Down Notification
When a remote manager is disabled, a Manager Down message box is displayed. You can perform one of the following actions:
• Click OK to manage all the backup containers and open submaps for those containers.

• Click Manage to manage all the backup containers but do not open submaps for those containers.

• Click Cancel to take no action.

**Responding To a Manager Up Notification**
When a remote manager is restored, a Manager Restored message box is displayed. You can perform one of the following actions:

• Click OK to take no action and close the message box. This gives the operator the opportunity to complete tasks being performed in the disabled manager’s submaps. When the tasks are completed, the submaps must be closed from the Navigation Tree or from the Manager Submap to make sure that the containers are automatically unmanaged.

• Select Close All to unmanage all the backup containers and close any submaps for those containers that are open.

**Changing Backup Default Actions**
You might want to change backup default actions to prevent notification of remote manager status changes and to automatically perform a specified action. You can change entries in the /usr/OV/app-defaults/Backup file or copy the appropriate entries in the user’s .Xdefaults file to change backup default actions.

The following lines in the /usr/OV/app-defaults/Backup file contain the default settings for the actions taken for remote manager status changes:

- backup*ManagerDown.displayPopup: TRUE
- backup*ManagerDown.defaultAction: OK
- backup*ManagerUp.displayPopup: TRUE
- backup*ManagerUp.defaultAction: OK

You can set both displayPopup values to FALSE to prevent notification of remote manager status changes. The values for ManagerDown.defaultAction (OK, Manage, or Cancel) and ManagerUp.defaultAction (OK or Close All) correspond to the actions initiated from the Manager Down and Manager Restored message boxes.

See “Responding To a Manager Down Notification” on page 190 and “Responding To a Manager Up Notification” for a description of those actions.

To automatically manage and unmanage backup containers without displaying notification of remote manager status, do the following:

• Change both displayPopup values to FALSE.

• Change the ManagerDown.defaultAction to Manage.

• Change the ManagerUp.defaultAction to Close All.

If you make changes to any of the app-defaults files after the Tivoli NetView product is started, use the command `xrdb -merge .Xdefaults` to load the new resource. If you have not started the Tivoli NetView product, the new resource will be loaded when you start the graphical interface.

**Determining Current Backup Sessions**
You can determine what containers a selected manager is managing on behalf of another manager using the Backup Session window. From the Backup Session
window, you can stop managing some or all of the backup containers, and you can open submaps for selected containers. To determine current backup sessions, follow these steps:

1. Select a manager symbol from either the Manager submap or another submap that contains a manager symbol.

2. In the context menu, click **Administer —> Backup —> Display Sessions.** The Backup Session Window is displayed.

3. Do one of the following:
   - Select specific containers from the list and select the **Close** button to stop managing some, but not all, of the containers in the list.
     The selected containers will be unmanaged, and any open submaps for those containers are closed.
   - Click **Close All** to stop managing all the containers in the list.
     All containers in the list will be unmanaged, and any open submaps for those containers are closed.
   - Select specific contains in the list and click **Open Submap** to open the submap for one or more containers in the list.
     Submaps are opened for those containers. If a submap is already open for any selected container, it is placed on top of the other windows that are displayed.
   - Click **Cancel** to exit the session window without performing any action.
Chapter 7. Managing Network Performance

The Tivoli NetView product helps you manage network performance by providing several ways to track and collect performance information for objects on the network. You can use performance information to help you:

- Monitor the network for signs of potential problems
- Resolve network problems
- Collect information for trend analysis
- Generate regular performance reports
- Allocate network resources
- Plan future resource acquisitions

This chapter describes MIBs and explains how to load, unload, and browse them. It also describes how to use Tivoli NetView predefined performance applications and how to create your own applications to monitor network performance. The following topics are covered in this chapter:

- “Loading and Unloading SNMP V1 MIBs”
- “Browsing SNMP V1 MIBs” on page 195
- “Using the Tivoli NetView Performance Applications” on page 197
- “Monitoring Real-Time Network Performance” on page 198
- “Collecting Historical Performance Information” on page 203
- “Monitoring File System and Paging Space” on page 209
- “Using the Tivoli NetView Graph Applications” on page 213
- “Generating Performance Reports” on page 215

Loading and Unloading SNMP V1 MIBs

You can use the Tivoli NetView product to work with either SNMP V1 MIBs or SNMP V2 MIBs. This section provides information about working with SNMP V1 MIBs.

**Note:** Use the Tivoli NetView Web console to work with SNMP V2 MIBs. Click **Tools —> Launch Web Console** to start the Web console. You can then use the Web console MIB Browser to work with SNMP V2 MIBs. See the **Tivoli NetView Release Notes**, Version 7.1 for more information.

The **Options —> Load/Unload SNMP V1 MIBs** operation lets you include your enterprise-specific MIB, or the enterprise-specific MIB for a device that you use on your network, in the Tivoli NetView product’s MIB description file. These menu operations work for SNMPv1 MIBs. You can also use these operations to load other MIBs.

The purpose of loading a MIB is to define the MIB objects so the Tivoli NetView applications can use those MIB definitions. The MIB you are interested in must be loaded on the system where you want to use **MIB Browser**, **MIB Data Collector**, **MIB Application Builder**, and the applications built by the MIB Application Builder. In a distributed network environment, load enterprise-specific MIBs on the Tivoli NetView server. Because the directory where the MIBs are stored, `/usr/OV/conf`, is automatically NFS mounted onto the client machines during the Tivoli NetView installation process, loaded MIBs are available to the clients. If you have unloaded all the MIBs in the MIB description file, you must load MIB-I or MIB-II before you can load any enterprise-specific MIBs.
Loading SNMP V1 MIBs

To load an enterprise-specific MIB, you can first copy the MIB into the default directory, /usr/OV/snmp_mibs. Otherwise, if you know the full path name of the directory where the MIB is located, you can enter it in the MIB File to Load text field in the Load MIB From File dialog box.

Each MIB that you load adds a subtree to the MIB tree structure. You must load MIBs in order of their interdependencies. A MIB is dependent on another MIB if its highest node is defined in the other MIB. For example, the MIB ibm-alert.mib is dependent on the MIB ibm.mib, so ibm.mib must be loaded before ibm-alert.mib is loaded.

Procedure for Loading SNMP V1 MIBs

Use the following procedure to load an SNMP V1 MIB:

1. In the Tivoli NetView main menu, click either Load/Unload SNMP V1 MIBs in the Options pull-down menu.

   The Load/Unload MIBs window is displayed, containing a scrollable list of all loaded MIBs, as shown in Figure 34.

2. Click Load to display the Load MIB From File dialog box, which contains a scrollable list of the MIB files in the default MIB directory, /usr/OV/snmp_mibs.

3. Select the MIB you want to load.

4. Click OK.

   The Load/Unload MIBs operation stores loaded SNMPv1 MIBs in the /usr/OV/conf/snmpmib database, which is known as the Loaded MIB Database. Do not try to edit this file directly; instead, make any changes to this file through the Load/Unload MIBs operation.

When the MIB is loaded, you can traverse the MIB tree and select objects from the enterprise-specific MIB to use in the following operations:

- Browsing MIBs
- Collecting MIB data
- Building MIB applications
- Running applications you build with the MIB Application Builder
Procedure for Unloading SNMP V1 MIBs

To unload enterprise-specific MIBs, select the MIB from the list of loaded MIBs in the MIB Load/Unload MIBs window and click **Unload**. Click **OK** in the Unload MIB—Confirmation dialog box to unload the MIB.

Browsing SNMP V1 MIBs

Use the MIB Browser to query and set MIB values for both Internet-standard and enterprise-specific MIB objects.

You can also use the MIB Browser to graph MIB objects and their specific instances. Some MIB objects can occur several times per network object, each time with a different value. Each such occurrence of the MIB object is called an *instance*.

For example, the interfaces MIB object ifDescription has as many instances as its associated network object has interfaces, as shown in the following example:

1: lo0; Software Loopback  
2: tk0; trty0; IBM 6611 Token-Ring Network Interface  
3: tkl; trty1; IBM 6611 Token-Ring Network Interface

However, the system MIB object sysContact has only one instance per network object, because generally there is only one person so designated. This is MIB instance 0, as shown in the following example:

sysContact.0: J. J. Wanscott 555-1234

To use the MIB Browser, follow these steps:

1. Select an SNMP network object whose MIB objects you want to view on a submap.
2. In the Tivoli NetView main menu, click **MIB Browser SNMP V1** for SNMPv1 agents in the **Tools** pull-down menu. The Browse MIB window contains the name or IP address of the network object you selected, as shown in Figure 35 on page 196.
3. In the Community Name field, enter the community name of the agent that is running on the selected object. If left blank, the community name defaults to public.

   The community name functions as a password for different levels of access to MIB objects. For example, to retrieve the value of a MIB object, you need to know the community name that permits SNMP Get operations. Generally, this community name defaults to `public`. If you want to change the value of a MIB object, you must know the community name that permits SNMP Set operations.

4. Select nodes along the path of the MIB tree and select either Up Tree or Down Tree to traverse the tree. Generally, the MIB objects you will be working with are contained in either the `mgmt` branch, which contains standard MIB definitions, or the `private` branch, which contains enterprise-specific MIB definitions.

5. When you have reached a leaf node, or actual MIB object, select Describe to see a description of the object to determine whether it is the one you want.

   The Describe MIB Variable dialog box displays the name of the MIB object, its object ID, and the type of MIB object it is, and gives a brief description of its meaning.
Figure 36 shows the description of the snmpOutTraps MIB object.

<table>
<thead>
<tr>
<th>NAME</th>
<th>.iso.org.dod.internet.mgmt.mib-2.snmp.snmpOutTraps</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJECT ID</td>
<td>.1.3.6.1.2.1.11.22</td>
</tr>
<tr>
<td>TYPE</td>
<td>Counter</td>
</tr>
<tr>
<td>ACCESS</td>
<td>Read-Only</td>
</tr>
</tbody>
</table>

**DESCRIPTION**

The total number of SNMP Trap PDUs which have been generated by the SNMP protocol entity.

6. To query the MIB object for its current value, click **Start Query**.
   You do not have to be at a leaf node in order to use the Start Query operation. You can begin a query at any intermediate node to retrieve all the MIB objects in that node's subtree at the same time. Once you see the list displayed in the MIB Values field, you can select one and look at its description. The current value of the MIB objects will be displayed in the MIB values fields. To stop the query, click **Stop Query**.

7. You can graph real-time data on the MIB object you have selected, provided it is numeric. Click **Graph** to display the Tivoli NetView Grapher window and look at the real-time values of the selected MIB object.

**Using the Tivoli NetView Performance Applications**

The Tivoli NetView product provides applications that enable you to monitor both real-time and historical network performance. The predefined applications monitor and graph real-time network performance. These predefined applications are described in “Using Tivoli NetView Predefined Applications” on page 198. The MIB Application Builder enables you to create your own applications to collect, display, and save real-time MIB data.

The MIB Data Collector provides a way to collect and analyze historical MIB data over long periods of time to give you a more complete picture of your network’s performance.

The MIB Application Builder is described in “Building MIB Applications” on page 201.

**Comparison of MIB Applications**

Table 18 on page 198 compares the MIB Application Builder and the MIB Data Collector.
### Monitoring Real-Time Network Performance

The Tivoli NetView product provides applications that enable you to monitor network real-time performance. It also provides a tool, the MIB Application Builder, that enables you to build applications. This section describes the Tivoli NetView predefined applications and the MIB Application Builder.

### Using Tivoli NetView Predefined Applications

Many of the performance applications described in this section display their data in graphs. Some performance applications do not graph the data they collect; see the individual application description for more information. Refer to the Help system entries for each application for complete information.

---

**Table 18. Comparison of Tivoli NetView MIB Applications**

<table>
<thead>
<tr>
<th>Task</th>
<th>MIB Application Builder</th>
<th>MIB Data Collector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting</td>
<td>Must always be started by clicking <strong>Tools —&gt; MIB Application Builder</strong> in the main menu</td>
<td>Data collection automatically started when the Tivoli NetView product is started, unless the collection is in suspended mode</td>
</tr>
</tbody>
</table>
| Selecting MIB Variables       | • Can select more than one MIB variable  
• Can apply selection rules to MIB variables | • Can collect data for one or all MIB instances  
• Can specify a single interface, but data is collected on the MIB index (instance) |
| Specifying Polling Intervals  | Can specify polling intervals only for Graph format                                         | Can change polling intervals for individual devices                                |
| Selecting Agents              | Agents must be selected from map                                                         | Define agents by using pattern-matching characters, selecting from map, or by entering the agent name in the field. |
| Setting Thresholds            | Not applicable                                                                           | Can choose one of the following:  
Exclude Collection  
Store, Check Thresholds  
Store, No Thresholds  
Do not Store, Check Thresholds |
| Generating Events             | Not applicable                                                                           | Can generate events and define specific trap numbers                             |
| Selecting Output Format       | Choose table, form, or graph                                                             | Shows data in table format. You can click **Graph** in the table display.          |
| Customizing Output            | • Can customize the “name” of the MIB variable displayed in the title bar  
• Cannot customize line names  
• Customization of graphic display lost between invocations | Uses applications in the /usr/OV/reports/C/ directory to graph output and generate reports |
| Saving Output                 | Can save output to /usr/tmp/xnmappmon.save or choose another file name                   | Collection file always stored in /usr/OV/database/snmpCollect. You cannot specify the file name |
| Exporting to ASCII Format     | • Cannot export graph data to ASCII format  
• Can export summary instance table format  
• No time points are provided (for **AIX, Solaris, and Linux**) | • Can convert to ASCII format  
• Time points provided through the Tivoli desktop (for **AIX, Solaris, and Linux**) |
| Performing Other Calculations on MIB Variables | Not applicable | Apply expressions in /usr/OV/conf/mibExpr.conf file to MIB variables |
Monitoring CPU Performance: If you suspect that some network nodes are handling more tasks than others and might become overloaded, you can check this by clicking System Activity —> CPU Performance in the Monitor pull-down menu. To use this application, one of the following conditions must be true:

- The trapgend daemon must be installed on a RS/6000
- The node must be an HP node.
- The Tivoli NetView MLM feature of the Tivoli NetView product must be installed.

This application monitors and graphs the average number of jobs performed in the last minute. Use this application to determine whether the system workload needs to be balanced.

Monitoring Disk Space: In the Monitor pull-down menu, click System Activity —> Disk Space to check the status of disk space on a remote SNMP agent node. This operation provides the following information:

- Name of the file system
- Total KB of space
- Number of KB used
- Number of KB available
- Percent of total capacity being used
- Directory name on which the file system is mounted

To use this application, one of the following conditions must be true:

- The trapgend daemon must be installed on a RS/6000 node.
- The node must be an HP node.
- The Tivoli NetView MLM feature of the Tivoli NetView product must be installed.

Monitoring Interface Traffic: If you suspect network performance problems, you can click Network Activity —> Interface: Traffic in the Monitor pull-down menu to see a graph of packet statistics for selected SNMP nodes. If the graph reveals that packets are not being received or transmitted by certain nodes, you can then begin fault management procedures for those nodes.

Monitoring Ethernet Performance: You can monitor and graph performance for Ethernet networks by using the operations listed under Network Activity —> Ethernet in the Monitor pull-down menu.

Monitoring Ethernet Traffic: In the Monitor pull-down menu, click Network Activity —> Ethernet —> Traffic to monitor and graph interface statistics for a Cisco gateway server that supports SNMP. The following information is displayed in the graph:

- The number of packets successfully transmitted by the interface hardware
- The number of packets successfully received by the interface hardware
- The number of bits per second successfully transmitted
- The number of bits per second successfully received

Monitoring Ethernet Errors: In the Monitor pull-down menu, click Network Activity —> Ethernet —> Errors to view interface error statistics for a gateway server that supports SNMP. This operation graphs the average number per second as computed over the previous polling interval for the following statistics:

- The number of packets received with a Cyclic Redundancy Checksum (CRC) value that is not valid
• The number of packets received with a Frame Check Sequence (FCS) value that is not valid
• The number of packets received that are smaller than the physical media permits
• The number of packets received that are larger than the physical media permits
• The number of packets that arrive too quickly for the interface hardware to receive.

**Monitoring TCP Connections:** In the **Monitor** pull-down menu, click **Network Activity —> TCP: Connections** to list the TCP connection table for selected remote SNMP nodes. The list shows the local and remote addresses of the node and the state of each session. This operation can help you determine whether performance problems are caused by the TCP layer instead of the SNMP layer.

**Monitoring SNMP Network Activity:** You can monitor and graph performance for SNMP networks by using the operations listed under **Network Activity —> SNMP** in the **Monitor** pull-down menu.

**Monitoring SNMP Traffic:** In the **Monitor** pull-down menu, click **Network Activity —> SNMP —> Traffic** to monitor and graph the SNMP network traffic to and from selected nodes. You can compare this graph to general interface traffic graphs to determine the percentage of traffic to and from the selected node that is attributed to the SNMP management of the selected node. The following information is displayed in the graph:

• The number of packets per second received by the selected node that carry SNMP traffic
• The number of packets per second transmitted by the selected node that carry SNMP traffic

**Monitoring SNMP Operations:** In the **Monitor** pull-down menu, click **Network Activity —> SNMP —> Operations** to monitor and graph the SNMP operations requested of and performed by the SNMP agent on the selected nodes. The number of operations reported is the number of operations for the given category that have occurred since the SNMP agent on the selected node was last started. The following information is displayed in the graph:

• The number of MIB objects retrieved successfully from the selected node through Get and Get Next requests
• The total number of MIB objects modified successfully on the selected node through Set requests
• The number of SNMP Get requests received and successfully processed by the selected node
• The number of SNMP Get Next requests received and successfully processed by the selected node
• The number of SNMP Set requests received and successfully processed by the selected node
• The number of SNMP Traps (unsolicited notifications) transmitted by the selected node

**Monitoring SNMP Errors:** In the **Monitor** pull-down menu, click **Network Activity —>SNMP —> Errors** to display a summary of the SNMP protocol errors detected by the SNMP Agent on the selected nodes. The number of errors reported is the total number of errors for the given category that have occurred since the SNMP Agent on the selected node was last started. The following information is displayed in the graph:
- ASN parsing errors
- SNMP requests by an unsupported version of SNMP
- SNMP requests received with an incorrect community name
- SNMP requests received with a community name that is inappropriate for the operation requested
- SNMP requests that could not be processed because the request message is larger than the maximum message size supported by the selected node
- SNMP requests that contained a MIB object not supported by the selected node
- SNMP set requests that contained values not valid for the specified MIB object
- SNMP requests that could not be processed for some other reason

**Monitoring SNMP Authentication Failures:** An authentication failure occurs when a management system sends an SNMP request to an agent but does not send the correct community name, or password, with the request.

If you have a map open with read-write access, you can obtain a list of the management systems that have caused authentication failures on a selected node. The node you select must be an HP node.

In the **Monitor** pull-down menu, click **Network Activity —> SNMP —> Authentication Failures**. From the resulting display, you can learn that:
- Certain management systems are not permitted to communicate with this particular agent.
- The agent might have a configuration problem, which you can take steps to resolve.

**Building MIB Applications**
The MIB Application Builder enables you to build MIB applications without programming. This is helpful when you want to monitor real-time performance of specific MIB objects. You can build MIB applications that poll certain MIB objects on a regular basis and produce output such as forms, tables, or graphs.

The new MIB application is placed in the Monitor menu by default. To look at MIB values returned by the new application, select the appropriate item under the Monitor option on the Tivoli NetView main menu.

In the **Tools** pull-down menu, click **MIB Application Builder**. The MIB Application Builder dialog box is displayed, as shown in Figure 37 on page 202.
The MIB Application Builder window contains a list of all existing MIB applications and the location of their selections in the Tivoli NetView menu structure. You can select an existing application to modify or delete, or you can add a new application.

**Adding a MIB Application:** To add a MIB application, click the **Add** button. The Add MIB Application dialog box contains four sections, each of which is described in the following paragraphs.

The top section contains text fields where you can enter an Application ID, or name, for your MIB application. This name should be the same as the name you specify in the menu path that enables users to access your application. Your MIB application will be available from the Tivoli NetView main menu and from the object context menu.

The Application Title field enables you to specify the text that is displayed on the title bar in the application once it has been selected. You can select one of the following formats for your application's output:

- **Form:** Use only MIB objects that have a single instance per system, that is, those that occur only once.
- **Table:** No restrictions identified.
- **Graph:** Use only MIB object types Integer, Counter, and Gauge. Variables with these types can have more than one instance per system.

The Display Fields section contains data about the MIB objects you select to be monitored by this application. Click the **Add** button to display the MIB
Application Builder/Add MIB Objects dialog box. Use this dialog box the same way you used the Browse MIB dialog box to find the MIB objects you want to add to your application.

The Tivoli NetView Integration section is where you decide what menu path users must traverse to select your application. The default main menu selection is Monitor; you can choose selections under Monitor that are most appropriate for your application, or add selections if necessary. You can also change the selection rule, which directs the application to begin only if the selected objects meet the criteria found in the rule. For example, if the selection rule is:

(isSNMPSupported || isSNMPProxied)

The MIB application will run only if the objects selected on the map support SNMP directly or through a proxy agent. If you select an object that does not meet the selection criteria, the MIB application’s menu selection is grayed.

In the bottom section, Help Text, you can enter some basic information about what your application does and how to use it. It is good practice to enter at least a minimum amount of information in case someone wants to know what your application does. A user can access Help for your application by clicking Help on the application menu bar and selecting On Application.

Once you have completed all sections of the Add MIB Application dialog box, click OK to add the new MIB application to the list of available MIB applications. Click Close to exit from the MIB Application Builder.

Running MIB Applications through the Graphical Interface: To run the application, select an object or objects on the network map, click the Monitor option in the Tivoli NetView main menu, and follow the menu path you entered for your application. If your application graphs MIB values, it runs in the Control Desk window until you stop it by clicking File —> Exit in the application menu bar. If your application produces a table or form, it runs in a separate window.

If a MIB application does not run, check the following conditions:
- Ensure that the MIB is loaded into the Loaded MIB Database. Sometimes, a MIB is loaded for the specific purpose of creating a MIB application, and then it is unloaded to conserve space in the database.
- Check that the MIB object ID you have selected is supported by the device you want to monitor.
- Ensure that the application was created to monitor the selected device and that the community name is properly set. Check the /usr/OV/conf/snmp.conf file to verify the community name.

Running MIB Applications Using the runapp Command: You can run an existing MIB application without using the MIB Application Builder. The runapp command executes the MIB application, using the application name and host name you specify, as shown in the following example:

runapp -a mib0bjApp -h host1

For more information about the runapp command, refer to the man page.

Collecting Historical Performance Information

You can compile historical performance data about your network by using the Tivoli NetView MIB Data Collector. This tool enables you to manipulate data in several ways, including:
• Collect MIB data from network nodes at regular intervals.
• Store MIB data in a file. If the Tivoli NetView product is configured to work with a relational database, you can transfer collected data to a relational database and use the relational database tools to create reports. Refer to the Tivoli NetView for UNIX Database Guide for more information about transferring collected data to a relational database.
• Define thresholds for MIB data and generate events when the specified thresholds are exceeded. Setting MIB thresholds enables you to automatically monitor important network and system parameters to help you detect and isolate problems.

Using the MIB Data Collector
Before you configure your system to collect MIB data, you need to understand the definitions of the MIB objects and what they do. To look at descriptions for selected MIB objects, click MIB Browser: SNMP in the Tools pull-down menu. Refer to the vendor documentation for information about their enterprise-specific MIBs.

Also ensure that there is enough room to store data in the /usr/OV/databases/snmpCollect directory. It might be necessary to remove some files in this directory to make space available. You can set a crontab command to periodically remove files, or use the snmpColDump command to edit the directory files and delete selected lines. See the man pages for more information about using these commands.

In the Tools pull-down menu, click Data Collection & Thresholds: SNMP. The MIB Data Collection dialog box is displayed, as shown in Figure 38 on page 205.
The MIB Data Collection dialog box contains three sections:

- **MIB Objects Configured for Collection**
  
  Use this section to select the MIB objects or MIB expressions on which to collect data.

- **MIB Object Collection Summary**
  
  Use this section to identify the nodes from which to collect MIB data and to view the current collection defaults.

- **Collection Details**
  
  Use this section to identify details about the data collection such as how often to collect data and whether to send threshold events.

**MIB Objects Configured for Collection**:
The MIB Objects Configured for Collection section of this dialog box contains the following fields:

---

![Figure 38: MIB Data Collection Dialog Box](image)

---
Status Indicates whether data collection is suspended or active. You can change the status by clicking either Suspend or Resume.

Label Displays the name of the file to which data will be stored.

MIB Object ID Displays the object identifier of the MIB object configured for collection. If the value in this field does not begin with a dot (.), it is a MIB expression. This is the path through the MIB tree that ends in this particular leaf node, or object.

MIB Object Collection Summary: The MIB Object Collection Summary section of this dialog box contains the following fields:

Interval Specifies how often data is collected from a source. The s, m, h, or w after the number indicates seconds, minutes, hours, or weeks.

Store Indicates whether or not data is being stored.

Threshold Displays two values: the first is the threshold, the second is the rearm. If the threshold and rearm values are not defined, the Threshold column displays 0.00 0.00.

Source Displays the name of the collection source. The source name can be an individual node or a set of nodes based on an IP address wildcard, for example, 15.212.*.*.

Instances Displays the MIB instances on which data is to be collected. The instance is an internal counter on the system. For example, assume you have multiple disks on a node and the name of the MIB object is disks. The instance tells the data collector on which disk to collect data. An instance value that is an asterisk (.*)) signifies that data is to be collected on all instances. A numerical instance value specifies a particular instance. An instance can also be a regular expression.

Collection Details: The Collection Details section of this dialog box contains the following fields:

Collection Mode Determines the collection mode. Use one of the following four values:
• Exclude Collection
• Store, Check Thresholds
• Store, No Thresholds
• Do not Store, Check Thresholds

Polling Interval Determines how often data is collected from a source. Enter a positive real number followed by an s, m, h, or w that indicates seconds, minutes, hours, or weeks. If you do not enter a letter following the number, the data collector uses the default of seconds. For example, 1.5h indicates one and a half hours, and 30 indicates 30 seconds.

Trap Number Specify an enterprise-specific trap number. The default trap is 58720263, the enterprise-specific trap that the MIB Data Collector sends when a threshold or a rearm value is exceeded. The MIB Data Collector sends the trap using the enterprise ID of the management station.
Threshold
Enter a threshold value that specifies when you want to be notified of traffic patterns that are outside the normal expectations. When the threshold value is passed, the specified threshold event is generated.

Rearm
Enter an appropriate rearm value to control the frequency of threshold events generated. When a MIB value drops below or is equal to the rearm value, a rearm event is generated. Another threshold event will not be generated until the rearm event occurs and the collected value again exceeds the threshold value after being rearmed.

Instances
Use the option list to select the type of instances you want to enter in the text field to the right of the option button. Specify the instance of the MIB object on which you want to collect data. If the object on which you want to collect data does not support multiple instances, the instance is zero. If you have multiple instances of a MIB object on a node, you must specify the instance on which you want to collect data. See the dialog box help for more information.

Example of Collecting MIB Data
Suppose you want to collect information about the number of inbound SNMP packets received by objects on a particular network. Follow these steps:

1. In the Tools pull-down menu, click Data Collection & Thresholds: SNMP.
2. Select the snmpInPkts item from the default list of configured collections. The current status of snmpInPkts data collection is Suspended. Note the information about this list item that is displayed in the MIB Object Collection Summary area of the dialog box. In this area, you can click Add if you want to add other nodes from which to collect this data.
3. Select the list item in the MIB Object Collection Summary area of the MIB Data Collection dialog box. The bottom area of the dialog box, Collection Details, becomes active and displays the values from the Collection Summary area in fields that you can change.

The following values apply to this particular collection:
- The polling interval is 1 hour.
- The data collected is to be stored and checked against a threshold value.
- When the collection frequency exceeds 15 occurrences per hour, a threshold event will be generated, sending trap number 58720263 to the manager.
- Once the trap has been sent, the rearm value, 70%, controls the frequency with which subsequent traps signifying a threshold event will be sent.
4. If this information suits your needs, you can click Resume and then Apply to apply the change in status.

Notes:
- Although the Status field for snmpInPkts changes to Collecting after you click Resume, the change does not take effect until you click Apply at the bottom of the dialog box.
- Data collection is restarted each time you click Apply which means that polling is interrupted. For example, suppose that you have set polling intervals on a MIB object to 1 hour. Clicking Apply causes the data collector to begin polling, even if the last poll took place only 5 minutes ago. Clicking Apply several times will increase network traffic and store more data than you might have intended.
5. To see the collected data, click Show Data in the MIB Objects Configured for Collection section of the MIB Data Collection dialog box. Collected data will be...
displayed after the first polling interval you specified has passed. In this example, no data will be displayed until one hour has passed from the time you clicked **Apply** to resume data collection for snmpInPkts.

The data is displayed as a table that lists the polling interval, time of collection, the source node from which the data was collected, and the value of snmpInPkts for that node. If you want to see a graph of the collected data, click **Graph** at the bottom of the MIB Data Collection/Show Data dialog box.

6. You can exit from the **Tools —> Data Collection & Thresholds: SNMP** operation and view the collected data at a later time by selecting **Tools —> Graph Collected Data: SNMP** from the Tivoli NetView main menu. Again, you must wait until after the specified polling interval has passed before there will be any collected data to view.

If no data is being collected, make sure that the [snmpCollect daemon](#) is running on the manager. This daemon stops running when file system space is not available. If you have root authority, you can use the Tivoli desktop to restart the daemon. Then, you can restart the data collection.

If you are using a standalone Tivoli NetView installation or you are not using the Tivoli Management Framework product, you can use Server Setup function (click **Configure —> Set options for daemons —> Set options for event and trap processing daemons**) or you can use the following command:

```
ovstart snmpCollect
```

Refer to the man page for more information about the `snmpCollect` daemon.

**Graphing MIB Variable Expressions**

You can graph the result of expressions applied to MIB variables. Expressions for manipulating MIB variables are stored in the `/usr/OV/conf/mibExpr.conf` file. These expressions are in postfix format. You might want to try using the example expressions before you create others.

**Steps:** To create and graph the results of an expression, take the following steps:

1. Use one of the example expressions or add an entry to the `/usr/OV/conf/mibExpr.conf` file.
2. Select an object or objects on a subplot.
3. In the **Tools** pull-down menu, click **Data Collection & Thresholds: SNMP**.
4. Click **Add** in the MIB Data Collection dialog box.
5. Click **Expression** in the MIB Data Collection / MIB Object Selection dialog box.
6. Select an expression from the list that is displayed in the Expression ID text field and click **OK**. The MIB Data Collection / Add Collection dialog box is displayed.
7. Select the type of instances you will enter in the Instances text field to the right of the Instances option button. Or, enter `. *` to collect all instances of the selected MIB variables.
8. Click **Add From Map** to display the selected object or objects in the List of Collection Sources field.
9. To change the collection mode, click **Collection Mode**. The selections are as follows:
   - Exclude Collection
   - Store, Check Thresholds
   - Store, No Thresholds
• Do not Store, Check Thresholds

You can also change the threshold and rearm values if necessary. Click OK to close the dialog box.

10. The Status field in the MIB Objects Configured for Collection area of the MIB Data Collection dialog box now shows Collecting for this MIB object. However, you must click Apply to start the data collection.

11. Once data has been collected, you can display it by clicking Show Data in the MIB Data Collection dialog box. The data is displayed in table format, but you can convert it to a graph by clicking Graph at the bottom of the MIB Data Collection/Show Data dialog box. You can also graph the collected data at a later time by clicking Tools —> Graph Collected Data: SNMP in the Tivoli NetView main menu.

Using the setthresh Command
You can set a threshold for MIB data collection without using the Tools —> MIB Data Collection operation. The `setthresh` command sets up data collection configurations and stores the values in the /usr/OV/conf/snmpCol.conf file. If any entry already exists with matching MIB object ID and source name, that entry is updated with the new information.

Example of Collecting Thresholds: You can collect data or monitor thresholds only on numeric MIB variables, that is, those that are defined as type Counter, Gauge, or Integer. The following example shows how the `setthresh` command is used to collect the same data that was collected in the previous example:

```
setthresh -o snmp.snmpInPkts -s R -n *.s.*
-c W -m s -p 1h
-v 10 -r 70 -t %
-i ALL -T 58720263
```

For more information about the `setthresh` command, refer to the man page.

Monitoring File System and Paging Space

Collecting network and performance data can quickly deplete key system resources. If file system or paging space become full, some processes might stop. However, it is not always convenient to continually monitor the system for this problem.

When you want to monitor file system or paging space, click Local Filesystem and Paging Space in the Monitor pull-down menu to receive a trap or a message that informs you when a threshold condition has reached its limit. This option monitors the root file system and the paging space on the local manager system where Tivoli NetView is installed (usually /usr/OV). If /usr/OV/databases and /usr/OV/log are defined as separate file systems, the program also monitors them.

The file system and paging space monitoring process uses the values set in the Monitor File System & Paging Space dialog box.

The Monitor File System & Paging Space Dialog Box
When you click Local Filesystem and Paging Space in the Monitor pull-down menu, the Monitor File System & Paging Space dialog box is displayed as shown in Figure 39 on page 210.
You can start file system monitoring (with or without dynamic increase of file system space) and accept or change the default values for file system and paging space thresholds, the polling interval, and the notification method. The Monitor File System & Paging Space dialog box contains the following settings:

- **File System & Paging Space Monitor Status**
  - Monitor With Dynamic Increase
  - Monitor Without Dynamic Increase
  - No Monitor

- **File System Threshold (% used):** [97]
- **Paging Space Threshold (% used):** [87]
- **Polling Interval (seconds):** [300]

- **Notification:**
  - Message Windows
  - Events

**Messages:**

![Monitor File System & Paging Space Dialog Box](image)

*Figure 39. Monitor File System & Paging Space Dialog Box*

You can start file system monitoring (with or without dynamic increase of file system space) and accept or change the default values for file system and paging space thresholds, the polling interval, and the notification method. The Monitor File System & Paging Space dialog box contains the following settings:

- **File System & Paging Space Monitor Status**
  - Click **Monitor with Dynamic Increase** if you want to start file system and paging space monitoring and if you want file system size or paging space to be increased automatically when the threshold is reached.

  File system size is increased by 4 MB, and paging space is increased as follows:

  - If only one page space is defined and there are free partitions on the hard disk, the paging space is increased by 12 MB.
  - If more than one page space is defined and the page space with the smallest percentage of use has reached the threshold, the page space defined on the hard disk with the most free partitions is increased by 12 MB.
  - If more than one page space is defined on a single hard disk, then the page space with the smallest percentage of use is increased by 12 MB.
If no free partitions exist, users are notified by the notification method you specify (a message window or a trap).

The monitoring process also logs a message to the /usr/OV/log/shpmon.log file, indicating when the process was started, stopped, and the amount of increase that was made to the file system space or paging space.

- Click **Monitor Without Dynamic Increase** to start file system and paging space monitoring and to notify users when file system or paging space needs to be increased.

For Solaris and Linux:

- Click **Monitor without Dynamic Increase** to start file system and paging space monitoring and to notify users when file system or paging space needs to be increased.
- Click **No Monitor** when you want to stop the monitoring process. This is the default.

The monitoring process also logs a message to the /usr/OV/log/shpmon.log file, indicating when the process was started and stopped.

- **File System Threshold**
  The default threshold for file system space is 97 percent. You can use this value, or you can type a different value in this field. Users are notified when the threshold is reached.

- **Paging Space Threshold**
  The default threshold for paging space is 87 percent. You can use this value, or you can type a different value in this field. Users are notified when the threshold is reached.

- **Polling Interval**
  The default polling interval is 300 seconds. You can use this value, or you can type a different value in this field.

- **Notification**
  You can notify users with an audible message box or by generating a trap. The default is to generate a trap. You can configure the trap to be displayed and logged using the **Options —> Event Configuration —> Trap Customization: SNMP** operation.

### Starting and Stopping the Monitoring Process

To start or stop the file system and paging space monitoring process, follow these steps:

1. In the **Monitor** pull-down menu, click **Local Filesystem and Paging Space**.

   The Monitor File System & Paging Space dialog box is displayed as shown in Figure 39 on page 210.

   **Note:** For **Solaris**, the Monitor File System & Paging Space dialog box is slightly different than Figure 39 on page 210.

2. Make the appropriate changes to the dialog box. See “The Monitor File System & Paging Space Dialog Box” on page 209 if you need more information about the dialog box settings.

3. Click **Apply** to accept the change.

4. Click **Cancel** to close the dialog box.
Note: If you stop the monitoring process when the process is inactive (depending on the polling interval), it seems as though the process is still running. Because the termination signal has not yet been received, if you attempt to start the monitoring process again, a message is displayed telling you that the process is already running. When the monitoring process becomes active, the monitoring process receives the termination signal and stops.

Starting from the Command Line

You can start the file system and paging space monitoring process from the command line and specify what options to use. Use the following command to start the monitoring process:

For Enter

AIX  `shpmon [-m 1|0] [-e] [-t update_time][-Q]`

Solaris and Linux

  `shpmon [-e] [-t update_time] [-Q]`

Where:

-m 1|0

  Starts with or without dynamic increase. The value 1 starts monitoring with dynamic increase. You must be a root user to use this option. A 0 (zero) value starts monitoring without dynamic increase.

-e

 Executes the program one time and exits. The monitoring process checks the thresholds set in the Monitor File System & Paging Space dialog box and responds depending on what other flags are used. See “The Monitor File System & Paging Space Dialog Box” on page 209 for more information.

-t

  Specifies a polling time interval in seconds. If this option is not specified, the value set in the Monitor File System & Paging Space dialog box is used. See “The Monitor File System & Paging Space Dialog Box” on page 209 for more detailed information.

-Q

  Suppresses all messages sent to the user when the threshold value has reached its limit. Messages about changes to the file system and paging space sizes are logged in the shpmon.log file. Events are generated when threshold values are exceeded and cannot be increased.

Monitoring Specific Events

To configure the Local Filesystem and Paging Space option to execute only when a specific event occurs, use the Event Configuration dialog box. When a threshold event occurs for the configured MIB object ID, the program executes, takes the appropriate actions, and exits.

Follow these steps to configure system events. You must be a root user.

1. In the **Tools** pull-down menu, click **Data Collection & Thresholds: SNMP**. On the **MIB Data Collection** dialog box, configure the threshold value and trap number for the disk utilization MIB object.

   See “Using the MIB Data Collector” on page 204 for those steps.

2. Click OK.

3. In the **Options** pull-down menu click **Event Configuration —> Trap Customization: SNMP**. On the **Event Configuration** dialog box, select the same trap number you configured on the MIB Collection dialog box. The trap
numbers are listed in the Event Identification section under Event. Use the Help option if you are not familiar with the Event Configuration dialog box.

4. Click Add. The Add Event dialog box is displayed.

5. Enter the following command in the Command for Automatic Action field:

   For Enter
   AIX, Solaris, and Linux
   /usr/OV/bin/shpmon -e -Q

6. Click OK to close the Add Event dialog box.

7. Click OK to close the Event Configuration dialog box.

This command executes the shpmon process only one time when the specified event occurs. It checks the file system and page space, increases them if necessary, and exits. Paging space is increased on the AIX operating system only.

---

Using the Tivoli NetView Graph Applications

Graph applications provide a convenient way to display performance information. This section describes how to use and customize graph applications.

Starting a Graph Application

To start a graph application, select an object or objects on a submap, then select the graph application whose results you want to view. All graph applications can be started from either the Tivoli NetView main menu or the object context menu. The graphs are displayed in a control desk window.

MIB Graph Applications

The following MIB graph applications can be started from an object context or main menu:

- Monitor —> System Activity —> CPU Performance
- Monitor —> System Activity —> Disk Space
- Monitor —> Network Activity —> Interface: Traffic
- Monitor —> Network Activity —> Ethernet —> Traffic
- Monitor —> Network Activity —> Ethernet —> Errors
- Monitor —> Network Activity —> TCP: Connections
- Monitor —> Network Activity —> SNMP —> Traffic
- Monitor —> Network Activity —> SNMP —> Operations
- Monitor —> Network Activity —> SNMP —> Errors
- Monitor —> Network Activity —> SNMP —> Authentication Failures

If you start one of these applications from the Tools window, you can place it in a control desk or drop it anywhere else on the desktop to make it a stand-alone application.

If you start one of these applications from an object context menu, it is displayed in the control desk. Starting one of these applications from an object context menu means it applies only to the selected object. If you want to collect and graph data for more than one network object, select the objects on the submap, then select the appropriate application from either the Tivoli NetView main menu or the Tools window.

Saving Performance Data

To save the performance data that is displayed in the graph application, click File —> Save As in the menu bar in the application window. You can save the data to
the default directory and file name, /tmp/xnmgraph.data, or specify a different
directory and file name. Click Apply to save the data. Look in the Messages area
for any messages generated as a result of clicking Apply. For example, if the save
is successful, the message tells you how many data points were saved for each
monitored MIB object.

The Save As operation dumps an ASCII file containing the data to the directory
and file name you specify. The ASCII file displays the collected data in table form
for each monitored MIB object.

To recreate the graph from the contents of the ASCII file, enter the following
command at the command line:

```
cat /u/<userid>/<directory>/<filename.ext> |
/usr/OV/bin/xnmgraph
-title "Monitor: SNMP Traffic"
-units "Packets / Sec"
-helpFile ovip/OVW/Functions/snmpTraffic
-mib "-1:Input Packets:::::::602route1.tivoli.com,-2:Output
Packets:::::::602route1.tivoli.com,"
```

**Printing Graphed Data**

To look at some of the graph information in printed format, print the contents of
the graph window. Select the Print Tool icon in the Tools window and drag it to
another area on the desktop to open a Print Tool window. Follow these steps to
capture and print the contents of the graph window:

1. Rearrange the windows on your desktop so that the window containing the
graph display does not overlap with any others.
2. In the Print Tool window, click Capture.
3. Move the pointer, which has changed shape from an arrow to a hand, to the
graph window you want to capture and click.
4. In the View Identification dialog box that is displayed, enter a file name for the
captured information. Click OK to close the View Identification dialog box.
5. The file name you specified is displayed in the Selected field of the Print Tool
window. You can either print the captured information immediately or save it
to print later.

Before you print, it might be helpful to save the captured information in a
directory in case you need to print it again.

6. To print the captured information immediately, select the name of the file and
specify the name of a printer and the number of copies, then click Print. The
Print Tool window closes.

7. To save the captured information, select the name of the file and then click
Save in the File pull-down menu in the Print Tool window. In the File Save
dialog box, enter the directory path where you want to store the file. You do
not need to enter a file extension because the Print Tool automatically adds .ps
to the file name if you save the captured image in a directory. Click OK.

8. To exit the Print Tool, click Exit in the File pull-down menu in the Print Tool
window.

Note that the Print Tool captures and prints only the information currently
contained in the graph window. It does not print the entire contents of the graph.
If you need to print the entire graph, you can page forward or backward and use
the Print Tool to print each page of graphed information. To page forward or
backward, click anywhere on the graph and click Page Forward or Page Backward
in the context menu that is displayed.
Showing Counter Values
If you are graphing MIB values of type Counter, you can click Show counters As in the View pull-down menu in the Print Tool window and choose one of the following ways to display the data:

Rate of Change
The default value shows the new counter value as a time-averaged value since the last query for the MIB object.

Actual Sampled Value
Contains the value returned from the MIB Counter variable.

Delta Value
Contains the change in the MIB variable since the last query for the MIB object. This value is not time-averaged.

Example
For example, suppose you are graphing a MIB variable with the following statistics:
Value of MIB variable at time 0 --> 100
Value of MIB variable at time 10 --> 300

The value from time 10 would be graphed in the following ways:

Rate of Change
20 (derived from (300−100)/10)

Actual Sampled Value
300

Delta Value
200 (derived from (300−100))

Adding a Line
An application sometimes has more lines to graph than the maximum number that are specified for the graph when the application is created. In the View pull-down menu, click Add Line to temporarily increase the number of lines the graph can display. This selection will be grayed out if the number of lines to be graphed does not exceed the maximum number of lines that can be graphed, as defined in the app-defaults file of the application.

Using the Context Menu
The context menu in a graph application enables you to zoom in and out so you can look at the graph from different perspectives. You can also use the context menu to page forward or backward through the collected data, or to display the beginning of the data, the end of the data, or all of the data.

Generating Performance Reports
The Tivoli NetView product provides sample shell scripts that use real-time and historical information to generate performance reports. In addition to using these shell scripts, you can create others that collect and display the performance information you most need to see. These reports are stored in the /usr/OV/reports/C directory.

In the Monitor pull-menu in the Tivoli NetView main window, click Reports: Site Provided. The sample shell scripts provided with the Tivoli NetView product are displayed in the Run Report File window.
Figure 40 shows the reports available with the Tivoli NetView product.

You can create your own reports to gather real-time or historical information about MIB objects. Your reports can be stored in the /usr/OV/reports/C directory, or in another directory of your choice. When you want to run a report that is stored in another directory, enter the full path name of the report in the Report File to Run field of the Run Report File dialog box.

Contents of the Report Directory

This section describes the files and shell scripts that are provided in the Tivoli NetView reports directory.

- The README file provides a description of each report in the /usr/OV/reports/C directory, including the type of object each report requires.
as a selection. It is good practice to update the README file each time you create a report and add it to the report directory.

- The graphDemo shell script provides an example of using the Tivoli NetView graph facility, called xnmgraph, to graph information that is collected in a standard file.
- The sumByDN shell script uses data from the SNMP Data Collector to provide an approximate estimate of which nodes have the most total traffic. This shell script must be run with data of type Counter.
- The graphSumByDN shell script uses data collected by the SNMP Data Collector to estimate which nodes have the most total traffic by time of day. Only nodes that are currently selected will be graphed.
- The NifInOctets shell script shows how to use the Tivoli NetView graph facility, xnmgraph, to graph a specific MIB object for selected nodes.
- The NifOutOctets shell script shows how to use the Tivoli NetView graph facility, xnmgraph, to graph a specific MIB object for selected nodes. The values must have been collected by the SNMP Data Collector.

**Writing Reports**

You can write reports that send their output directly to the terminal so you can see the results immediately. Getting immediate feedback can help you with monitoring and troubleshooting. You can also write reports that store their results in a file or in a database, where they can be used to chart performance trends over a period of time.

Some reports are run only if an object is selected from the map before the report is selected. Other reports require that more than one object be selected, or that no objects be selected. You can tell which reports require selected objects, and how many selected objects, by looking at the first character of the report name.

**Table 19. The number of Selected Objects a Report Requires**

<table>
<thead>
<tr>
<th>First Character of Report Name</th>
<th>Number of Selected Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>None</td>
</tr>
<tr>
<td>1</td>
<td>One</td>
</tr>
<tr>
<td>N</td>
<td>One or more</td>
</tr>
</tbody>
</table>

You might want to follow this convention when you create reports.
Chapter 8. Using the Agent Policy Manager (APM)

If you have the Tivoli NetView Mid-Level Manager (MLM) installed in your network, read this chapter to understand how to configure and to use the APM to more easily set up and view information about thresholds and file monitoring in a network.

This chapter is organized to accommodate both users who are familiar with Tivoli NetView MLM and those who are new to it and contains the following information:

- “What the APM Can Do for You”
- “Types of APM Policies” on page 221
- “Configuring and Starting the APM Daemon” on page 225
- “Starting the APM Configuration Application” on page 226
- “Creating, Changing, and Distributing an APM Policy” on page 227
- “Fixing an APM Policy” on page 233
- “Copying an APM Policy from One System to Another” on page 235
- “Completing the APM Policy Dialog Boxes” on page 235
- “Diagnosing Problems Using the Problem Determination Assistance Facility” on page 264
- “Example of Defining and Distributing a Policy” on page 269

This chapter also includes the “APM Reference” on page 272. This reference provides detailed information about how APM defines itself and its domains in the network, how community names are configured, and other reference information.

What the APM Can Do for You

If you use Tivoli NetView MLM in your network, you already know that it is a powerful application. Through the MLM, you can off-load some systems management from the Tivoli NetView product with the MLM’s thresholding capability. An MLM can identify a developing problem and alert you to it or take corrective action before your users ever become aware of it.

Tivoli NetView MLM is powerful, but it can require that you gain many new skills, especially if you are new to SNMP and the concepts of MIBs, community names, and sets and gets. APM, running in conjunction with Tivoli NetView MLM and the SmartSet Facility, allows you to accomplish these same tasks more easily by:

- Simplifying the task of defining threshold or file monitoring conditions by eliminating the need to define trap destinations.
- Providing the ability to distribute a threshold or file monitor configuration to groups of nodes as a single operation.
- Automatically creating submaps and an icon on your Tivoli NetView root map for MLMs and their managed nodes.
- Automatically creating icons on your root map that represent active threshold and file monitoring settings.
- Providing a way to filter file monitor traps so that only the ones you care about are forwarded to the Tivoli NetView Control Desk.

This management of agents in the network is accomplished by setting up policies. The policies you define are made up of two pieces:
- Defining rules about **which object** will be acted on (such as all routers, all machines in a building, or all devices in a certain subnet)
- Defining rules about **what** action will be taken (such as threshold on a MIB-II variable, monitor a log for an error message)

The process for adding a new policy is made up of two steps:

1. **Creating the new policy**
   - How to create the new policy is described in "Creating or Copying an APM Policy" on page 228.
2. **Distributing the policy** to the nodes in the SmartSet.
   - How to distribute the policy is described in "Distributing a Policy to Remote Nodes" on page 229.

APM simplifies the task of distributing changes to your network. Rather than NFS mounting Tivoli NetView MLM configuration files, or editing configurations on every MLM machine in your network, you use the SmartSet Facility to set up **SmartSets** to include all the nodes that will get a new policy. After you have defined the SmartSet to which the policy is to be distributed, it is a simple matter of clicking **Distribute** to update all the nodes in your network. The SmartSet Facility maintains a list of objects that fit SmartSet rules. It updates the list if changes in the network topology result in the addition of nodes that fit a SmartSet rule (or deletion of nodes already in a SmartSet). APM will act on these changes by configuring new nodes that fit the SmartSet rule or removing policies from nodes that no longer fit SmartSet rules. This policy maintenance takes place automatically.

See "Defining and Managing SmartSets" on page 83 for more information.

**APM SmartSet Icons That You Get Automatically**

APM and the SmartSet Facility create several new icons on the root map:

- **SmartSets** shows all SmartSets that have been defined. It is added to the root map by the SmartSet Facility, regardless of whether or not APM is configured to run.
  - Double-clicking this icon displays a submap showing all of the defined SmartSets. Initially, these SmartSets do not reflect status. When you double-click a SmartSet icon, status is set and maintained for that SmartSet.

- **MLM Managers** displays all the MLMs in your network, with all the IP nodes in each MLM’s domain.
  - Double-clicking this icon displays a submap of all MLMs in the network.
  - Double-clicking an MLM displays the MLM and the nodes in its domain.
  - Double-clicking a particular node shows details about the node itself, including IP interfaces and all distributed policies.

- **APM Monitors** is displayed after you define an APM policy. This icon indicates that APM policies are active.
  - Double-clicking this icon displays all the SmartSets to which policies have been distributed.
  - Double-clicking one of these SmartSets shows the nodes in the SmartSet against which the policies are set.
  - Double-clicking an individual node displays details about the node itself, including IP status and icons for the individual APM policies. The icons for APM policies are executable icons. Double-click on an icon to access the **Problem Determination Assistance** dialog box.
The Problem Determination Assistance dialog box is described in “Diagnosing Problems Using the Problem Determination Assistance Facility” on page 264.

**Propagation of Policy Status to Map Icons**

The color of each icon reflects the status of the policy as described in the following list:

- **Yellow**
  The APM Monitors icon on the root submap turns yellow when any threshold has been exceeded or when a file monitor condition has been met.

- **Green**
  Policy icons are green by default.

- **Blue**
  Policy icons turn blue when the distribution of the policy to the node failed. Threshold policy icons turn blue if the MLM session for thresholding the node goes down.

- **Red**
  File monitor policy icons turn red when a file monitor condition is met. Threshold policy icons turn red when arm conditions are met. Threshold policy icons turn green again when rearm conditions are met.

The MLM icon on the root map and its submaps propagates only IP status. The APM Monitors map aggregates IP status and APM policy distribution status. File monitor and threshold status are also aggregated into the APM Monitors map.

Because the SmartSets submap aggregates status, APM policy icon status also propagates to the SmartSets icon.

File monitor functions do not have a rearm definition. If the condition being monitored for is found and the APM Monitors icon changes color, reset the color manually through the Problem Determination Assistance dialog box.

The Problem Determination Assistance menu is described in “Diagnosing Problems Using the Problem Determination Assistance Facility” on page 264.

**Types of APM Policies**

This section describes the types of APM policies that you can define and distribute using the APM configuration application.

- “Threshold Policy”
- “File Monitor Policy” on page 222
- “Command Policy” on page 223
- “Filter Policy” on page 223
- “Trap Destination Policy” on page 224
- “Analysis Policy” on page 224
- “Alias Policy” on page 224
- “Administration Policy” on page 224

**Threshold Policy**

Through a threshold policy, you can collect important MIB data and can set thresholds to send a trap or to run a command when the threshold is tripped. Thresholds can be set on many types of MIB objects:

- MIB objects in the Tivoli NetView MLMMIB
• MIB-I or MIB-II objects
• MIB objects in standard or private MIB extensions that can be retrieved using SNMP gets (such as the MIB shipped with a router or other device)

With APM, you are setting a threshold against a group of nodes. The APM finds each of the nodes in the SmartSet and then determines which MLM in the network has monitoring responsibility for each of the nodes. It will be an MLM in the same domain as the node. The APM then does an SNMP set against the Threshold Table for each MLM to define the threshold. For example:
  You define a SmartSet MYCOLL that includes nodes A, B, C, and D.
  Nodes A and B are in the same domain as MLM 1.
  Nodes C and D are in the same domain as MLM 2.

Through APM, you define a threshold to be distributed to SmartSet MYCOLL. When you click Distribute, an SNMP set command is done against the Threshold Tables on MLM 1 and MLM 2. Each of these MLMs will begin polling their respective managed nodes for the MIB data specified in the threshold. MLM 1 now thresholds on nodes A and B, and MLM 2 thresholds on nodes C and D.

Note: If you are familiar with thresholding using the Tivoli NetView MLM configuration interface, you should note that APM differs from the MLM in that you cannot add an alias or node name to the beginning of the MIB object ID on which you are setting a threshold. Because you are setting thresholds against nodes in a SmartSet, there is no need for you to add a target to the front of the threshold variable. APM automatically creates the appropriate aliases and adds them to the beginning of the variable. APM also maintains those aliases so that they stay synchronized with the SmartSets and thresholds they represent.

Data about threshold and file monitor traps are recorded in files in the /usr/OV/databases/C5 directory for graphing by Problem Determination Assistance applications. See “Diagnosing Problems Using the Problem Determination Assistance Facility” on page 264 for more information.

**File Monitor Policy**

Through a file monitor policy, you can set up file and log monitoring on any node that has the Systems Monitor System Information Agent (SIA) installed. When you define a file monitor condition through the APM configuration application and distribute it, SNMP set commands are done against the actual SIA running on the nodes defined in the SmartSet for which you are defining a file monitor condition. Thus, the community names on the Tivoli NetView host where you are running the APM daemon and on the remote SIA nodes must be defined so that the management station can do SNMP sets on the SIA nodes.

Files of any type can be monitored, and you can monitor for many different conditions or combinations of conditions. Each condition returns a different trap. The conditions that you can monitor and the traps returned are:

<table>
<thead>
<tr>
<th>Condition Being Monitored</th>
<th>Trap Returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existence of a text string</td>
<td>Specific trap 21 - String found</td>
</tr>
<tr>
<td></td>
<td>Specific trap 22 - File data modified</td>
</tr>
<tr>
<td>Changes to characteristics of the file, such as owner, group, or permissions</td>
<td>Specific trap 23 - File status changed</td>
</tr>
<tr>
<td>Condition Being Monitored</td>
<td>Trap Returned</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>Existence of a file</td>
<td>Specific trap 24 - File does not exist</td>
</tr>
<tr>
<td></td>
<td>Specific trap 25 - File exists</td>
</tr>
</tbody>
</table>

See “Matched, Armed, and Disarmed Command Environment Variables” on page 275 for information about environment variables that correspond to MIB variables in the traps.

The Tivoli NetView product has a predefined filter that displays only Tivoli NetView MLM traps in the Control Desk. You create a dynamic workspace and select the predefined filter named C5filters to display the Tivoli NetView MLM traps only. Step 9 on page 271 in “Example of Defining and Distributing a Policy” on page 269 shows the steps for opening this dynamic workspace.

### Command Policy

Through a command policy, you can perform the following two types of command execution:

- Execute commands in the Korn shell environment. The results are placed in a MIB variable.
- Access kernel memory or a shared memory segment. The results are placed in a MIB variable.

You can execute any command and, optionally, return the results of the command in the form of a MIB variable. The results can be a display string, integer, counter, or gauge. An SNMP set or get to the variable results in the command being executed, and the output of the command is placed in the associated Result variable (depending on what type of output resulted from running the command).

A command policy also provides access to the kernel memory or a shared memory segment of a user application. Accessing the kernel or shared memory is much faster than executing a command and returning the result. With memory operations, the command table can set or get a value from a shared memory segment that an application has opened, or perform get operations from kernel memory. The results of a memory operation can be returned as a display string, integer, counter, or gauge.

Commands set up through a command policy can be run from either a set or get request. Set commands are done against the actual SIAs running on the nodes defined in the SmartSet for which you are defining a command policy.

### Filter Policy

Through a filter policy, you can define the action to be taken by the local MLM when certain traps are received. When a trap is received, the MLM compares the incoming trap information with the information in the filter policy. If the trap matches a filter policy, the action defined in that filter policy is taken.

If the incoming trap does not match filter policy, a defined default action is taken. The default action is defined in the Trap Destination table, which you can customize by defining a trap destination policy through APM. To control the protocol and the port that is used for trap reception, you must use the Tivoli NetView MLM configuration application to update the Trap Reception table.
If you want to reference an alias with a filter policy, create and distribute the alias policy separately.

**Trap Destination Policy**

Through a trap destination policy, you can specify the hosts that will receive traps from MLMs. A trap destination policy is used by the MLM to determine the destination for traps that fit one of the following criteria:

- The trap does not match filter policy and the defined default action is set to `sendTraps`.
- The trap matches a filter policy and the matched filter policy specifies to forward the trap, but there is no trap destination specified (the Destination field is blank).

APM automatically sets one APMHOST entry for each MLM to forward its traps to Tivoli NetView running APM. The trap destination is predefined to be all Tivoli NetView managers that discover MLMs that are thresholding and forwarding traps.

**Analysis Policy**

Through an analysis policy, you can evaluate complex mathematical expressions on local or remote MIB variables for the MLM. You can use the analysis result as the input for another operation (for example, as input to a threshold policy). Using an analysis result as input to a threshold policy, permits the data collection and verification of the final expression result. The expression is evaluated using floating point arithmetic and converted to the type of the final result.

If you want to reference an alias with an analysis policy, create and distribute the alias policy separately.

**Alias Policy**

Through an alias policy, you can associate a group of nodes with a single alias name, or a group of alias names with a single alias name. You can assign a node to multiple alias groups. This grouping feature enables you to monitor groups of nodes by specifying a single alias.

An alias policy enables you to define threshold, analysis, and filter policies for groups of similar nodes by using an alias for the group.

You will not see aliases that APM created (used for thresholding) in the list of aliases. Nor will you see aliases that are created by the Tivoli NetView product for the netmon daemon.

**Note:**

These aliases should not be changed manually. Only aliases that are created using APM alias policies will be in the list of aliases.

**Administration Policy**

Through an administration policy, you can store MLM administrative data in a tabular form. For example, you can use this policy to track available configurations, maintenance performed on a node, or the latest levels of software.
Configuring and Starting the APM Daemon

As shipped, the APM function is not automatically configured to run. To use APM, configure and start the APM daemon for the first time through the Server Setup application.

Configuring the APM Daemon through the Server Setup Application

To configure the APM daemon, follow this procedure:

1. Enter `serversetup` on the command line to access the Server Setup application.
2. Click `Configure —> Set options for daemons —> Set options for Agent Policy Manager daemon`.
   The Set options for Agent Policy Manager daemon dialog is displayed. By default, logging and tracing is on.
3. Make the appropriate changes in the Set options for Agent Policy Manager daemon dialog. Click the Help button on the dialog if you need more information.
   - Change the default log and trace files if desired.
   - Optionally, change the elapsed time between daemon attempts. This value determines how often APM will attempt to distribute threshold and file monitoring policies that it could not distribute previously (for example, if a node was down). By default, APM retries all failed distributions for all nodes once every 60 minutes. To turn off automatic retries, set this value to zero (0).
   - Optionally, change the number of threshold events stored in the history file. These events are used as plot points by the `xnmgraph` application. You can graph threshold history through the [Problem Determination Assistance application](#).
   - See "Diagnosing Problems Using the Problem Determination Assistance Facility" on page 264 for more information.
4. Click OK.

Now that the APM daemon is configured and registered, it starts automatically each time you start the Tivoli NetView product.

Starting the APM Daemon from the Command Line

After the APM daemon is configured, you can start it from the command line. The daemon is called `C5d`. Root authority is required to start the daemon. To start the daemon, enter the following command on the command line:

```
/usr/OV/bin/ovstart C5d
```

You can specify the following options when you start the `C5d` daemon:

- `-t tracefile`  
  Turns on tracing when the `C5d` daemon is started. Trace information is saved in the file specified.

- `-l logfile`  
  Log file is saved in the file specified.

- `-g`  
  Specifies the number of threshold events to be saved.

- `-r`  
  Specifies the retry interval.

See "Diagnosing Problems Using the Problem Determination Assistance Facility" on page 264 for more information.
When the daemon is running, you can use the -T and -L options to toggle tracing and logging, respectively, without having to stop and restart the daemon.

**Note:** The log and trace files can continue to grow without bounds. Periodically, delete or prune the files to prevent running out of file system space. By default, the log and trace files are stored in the `/usr/OV/log` directory.

See the C5d man page for more information.

## Starting the APM Configuration Application

You can start the APM configuration application in two ways:

- In the **Tools** pull-down menu, click **APM Configuration**.
- Double click **APM** on the Tools Window.

When you start APM, the Agent Policy Manager Configuration main menu is displayed as shown in Figure 41.

![Figure 41. Agent Policy Manager Configuration Main Menu](image)

From the Agent Policy Manager Configuration main menu, you can look at a list of current APM policies and their distribution status. "Distribution Status States" on page 227 describes the various status states.

By default, File Monitor policies are displayed. To switch to another type of APM policy, click **Policy Type**. You can add, delete, or modify a policy and distribute the policy to other nodes in the network. Much like the Tivoli NetView MLM Configuration Application, the APM configuration application gives you fields for defining configuration information about policies. You specify SmartSets to which a policy will be distributed. You can also view the distribution status of a selected policy to the individual nodes in the SmartSet.

Note that the first time you start APM, it might take some time before it is displayed because it must set status on all nodes in the associated SmartSets Agent Policy Manager uses and create submaps for the domain SmartSets. After the initial synchronization is not delayed.
You can use the Administer —> Start Application —> APMmap menu option to start the APM maps without starting the APM configuration application and without having to close and restart the Tivoli NetView graphical interface. You might find this useful if the APM configuration application ends or if you start the APM daemon after you have started the Tivoli NetView graphical interface. If you recycle the APM daemon, you will have to restart APM in this manner to update the maps. The same process that manages maps also displays the configuration application.

Distribution Status States

Table 20 describes the various policy status states.

<table>
<thead>
<tr>
<th>Status</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>NeverDistributed</td>
<td>An attempt has never been made to distribute this policy.</td>
</tr>
<tr>
<td>Distributed</td>
<td>The policy was successfully distributed to all nodes in the SmartSet.</td>
</tr>
<tr>
<td>PartiallyDistributed</td>
<td>An attempt was made to distribute the policy, but one or more nodes could not be modified. The nodes might not have been reached. If the failure is due to a timeout, you might want to let APM continue to redistribute on its own.</td>
</tr>
<tr>
<td>PartiallyDeleted</td>
<td>An attempt was made to delete the policy, but one or more nodes could not be reached to delete its definition. The nodes might not have been reached.</td>
</tr>
<tr>
<td>PendingModifyDistribute</td>
<td>A modification was made to the policy, but it has not yet been distributed.</td>
</tr>
<tr>
<td>PendingDeleteDistribute</td>
<td>The user selected a policy in the APM Configuration main menu and clicked Delete. The delete request has not been distributed to the SmartSets yet.</td>
</tr>
<tr>
<td>DistributeInProgress</td>
<td>APM was doing a distribution when the C5d daemon went down or a logic error occurred. If you see this error after C5d recycles, allow the C5d daemon to continue to distribute. If you are seeing several items with this status, recycle the daemon and call support.</td>
</tr>
<tr>
<td>ModifyDistributeInProgress</td>
<td>APM was distributing a modified policy when the C5d daemon stopped or a logic error occurred. If you see this error after C5d recycles, allow the C5d daemon to continue to distribute. If several items have this status, recycle the daemon and call support.</td>
</tr>
<tr>
<td>DeleteDistributeInProgress</td>
<td>APM was distributing a deletion when the C5d daemon stopped or a logic error occurred. If you see this error after C5d recycles, allow the C5d daemon to continue to distribute. If several items have with this status, recycle the daemon and call support.</td>
</tr>
</tbody>
</table>

Creating, Changing, and Distributing an APM Policy

This section describes the following tasks which you can perform from the Agent Policy Manager Configuration main menu:

- “Creating or Copying an APM Policy” on page 228
- “Viewing Policy Details” on page 229
- “Distributing a Policy to Remote Nodes” on page 229
- “Managing Policy Distribution Using the Node Distribution Status Dialog Box” on page 230
- “Modifying an Existing Policy” on page 231
- “Viewing Pending Policies” on page 232
- “Deleting an Existing APM Policy” on page 232
- “Undoing a Change” on page 233
Creating or Copying an APM Policy

You can create a new APM policy or you can base a new APM policy on an existing policy. If you are creating a file monitor or threshold policy, it is best to use the default policies provided as the basis for your new policy. If you create a new file monitor or threshold policy, and you leave some fields blank on the Threshold or File Monitor dialog, you might get unexpected results, such as not receiving traps or the color of icons not changing.

To create a new APM policy, follow this procedure:

1. In the Agent Policy Manager Configuration main menu, click Policy Type. By default, File Monitor is selected as the type of policy that is to be created.
   A list of currently defined policies is displayed.
2. Do one of the following:
   - To create a new policy, click Add/Copy without selecting a policy.
   - To base a new policy on one that you have already defined, select the policy you want to copy, and click Add/Copy.
     If you are creating a file monitor or threshold policy, you should use the default policies that are provided to ensure that all fields are filled in. This ensures that traps will be returned correctly.

   The dialog box for defining the policy type you selected is displayed.
3. Fill in the fields in the dialog box.
   Note that the new policy you are creating must have a unique name across all APM policies. In other words, you cannot have two thresholds or two file monitor policies called Test, nor could you have one threshold policy called Test and one filter policy called Test. If you are creating a new file monitor or threshold policy, make sure you fill in the Specific Trap and Arm Enterprise fields in the Threshold Actions dialog box and the Rearm Specific Trap and Rearm Enterprise fields in the Rearm Actions dialog box if you have a rearm condition. Use the values from the default policies as follows:
   - In the Threshold Actions dialog box, specify 1 in the Specific Trap field and .1.3.6.1.4.1.2.6.12.5.1 in the Arm Enterprise field.
   - In the Rearm Actions dialog box, specify 2 in the Rearm Specific Trap field and .1.3.6.1.4.1.2.6.12.5.1 in the Rearm Enterprise field.

   See "Completing the APM Policy Dialog Boxes" on page 235 for additional information about dialog box fields. If you are new to Tivoli NetView MLM, refer to the Tivoli NetView MLM User’s Guide.
4. Specify the SmartSets to which the policy is to be distributed clicking Assign in the dialog box.
   The SmartSet Assignments dialog box is displayed.
5. Select the SmartSets on the SmartSet Assignment dialog box, click Assign, and then click OK.
   Note that if you do not have the SmartSet defined before bringing up the APM configuration application, you can start the SmartSet Editor from the SmartSet Assignments dialog box.
   The SmartSet Assignments dialog box closes.
6. Click Apply in the policy dialog box.
   You will see a message indicating that the policy was saved and that you can distribute it from the Agent Policy Manager Configuration main menu.
   See "Distributing a Policy to Remote Nodes" on page 229 for more information.
Viewing Policy Details

To view a policy, select the appropriate policy from the Agent Policy Manager Configuration main menu and click View Current. The policy dialog box, which you completed when you created or modified the policy, is displayed and contains the details of the selected policy.

Distributing a Policy to Remote Nodes

After you have successfully created a policy, you can distribute it to the nodes specified in the SmartSet assigned to the policy. You can distribute a policy from the APM main menu in two ways:

- **To all nodes** in the SmartSet. Use this method when you distribute a policy or change for the first time.
- **To individual nodes** in the SmartSet. Use this method if you have already distributed a policy or change and you want to manually retry the distribution to selected nodes.

For thresholds, the distribution process consists of SNMP set commands on the specified MIB object on the MLMs assigned to nodes in the SmartSets. Thus, you must have the community names on the node where you are running APM and the target MLMs set up so that the APM node is allowed to do SNMP sets on the target nodes.

Note that you do not have to redistribute a policy if nodes have been added or deleted in target SmartSets or if you have modified the SmartSet rule in some way. The information is automatically redistributed by APM. You also do not have to redistribute to a failed node. APM retries distribution on a regular interval, by default, every hour.

Distributing a Policy to All Nodes

From the Agent Policy Manager Configuration main menu, select the policy you want to distribute by clicking it, and then Distribute. The dialog that is displayed lists the name of the policy and only those nodes to which the policy is to be distributed. For policies being distributed to MLMs, the target SmartSets resolve to a list of nodes. Click Start to distribute the policy. Messages about the success or failure of the distribution appear in the Messages area on the Distribute dialog box.

Distributing a Policy to Individual Nodes

To distribute a policy to individual nodes, you must have distributed the policy to all nodes at least once. Then you can distribute the policy to selected failed nodes. To distribute a policy to individual nodes, follow this procedure:

1. From the Agent Policy Manager Configuration main menu, select the policy you want to distribute by clicking it, and then Node Status.

   The Node Distribution Status dialog box is displayed.

2. Select the nodes to which you want to distribute the policy, and then click Distribute.

   The dialog box that is displayed lists the selected target nodes.

3. Click Start to distribute the policy.

   Messages about the success or failure of the distribution appear in the Messages area on the Distribute dialog box. APM distributes the policy to only the selected nodes and does not automatically retry failed distributions to nodes that are not selected.
Successful Distribution
If all nodes in the SmartSet successfully receive the policy, the policy status in the
Agent Policy Manager Configuration main menu is changed to Distributed to
indicate that the policy was distributed.

Distribution Failures
If any or all of the distributions fail, the policy status is changed to
PartiallyDistributed on the Agent Policy Manager Configuration main menu.
Failures can occur for several reasons:

- An incorrect community name (causes an authentication error). See “Configuring
Community Names” on page 272 for more information.
- A network failure.
- An SNMP agent or application not responding.
- The target MIB is not loaded.
- The distribution attempt times out.
- The node is no longer present in the Tivoli NetView product and not yet deleted
from the topology database.

If you receive a message that targets are not defined, it indicates one of two
conditions:
- On a file monitor policy, the target SmartSet is empty.
- On a threshold policy, APM either could not find an MLM in the network whose
domain contains any of the nodes in the SmartSet, or it could not find any
MLMs.

You can continue to attempt to redistribute policies manually, or you can let APM
continue to try to redistribute. APM continues to periodically retry the distribution
in the following ways:
1. Periodically, depending on the interval set when the C5d daemon was started.
   This option can be set using the -r option. All failed distributions are tried at
   the same time. The default interval is to retry distribution once an hour. To stop
   retries, use the -r option with zero (0) as the retry interval (-r0).
2. When a policy is redistributed (either manually by a user or automatically
   when a SmartSet is changed)
3. When a previously unresponsive node responds to a set request for another
   policy. When the node responds, APM checks to see if there are any other
   outstanding distributions for that node and attempts to redistribute any it
   finds.

Managing Policy Distribution Using the Node Distribution
Status Dialog Box
After you have distributed a policy, you can use the Node Distribution Status
dialog box to perform the following tasks:

- View the distribution status to the individual nodes.
- Distribute the policy to failed nodes. See “Distributing a Policy to Individual
  Nodes” on page 229 for these steps.
- Delete an obsolete node from the distribution list.
- Reset all the APM policies to selected nodes.
To display the Node Distribution Status dialog box, select the appropriate policy on the Agent Policy Manager Configuration main menu and then click **Node Status**. The Node Status dialog box is displayed as shown in [Figure 42](#).

![Figure 42. Node Distribution Status Dialog Box](image)

The Node Distribution Status dialog box lists the target nodes, the target MLM and the policy distribution status (whether the add or delete failed or was successful). If the selected policy is a threshold policy, the dialog box also lists the MLM that is doing the thresholding on behalf of the target node.

You can select one or more nodes and then click one of the following buttons:

**Distribute**

Retry distribution of just this policy to the selected nodes. See "Distributing a Policy to Individual Nodes" on page 229 for more detailed information.

**Delete**

Delete the selected obsolete nodes from the distribution list. An obsolete node is a node that has been previously distributed to but is no longer a member of any target SmartSet for the policy. When a node is removed from the target SmartSet, APM automatically tries to delete the policy from the node. If the delete operation fails, the node remains in Delete Failed status until the delete is successful, the node is deleted from the ovwdb database, or you manually delete the node from the distribution list using this function. If you try to delete a node that is still in a target SmartSet, an error message is displayed.

**Resynch**

Reset all the APM policies for the selected nodes. The Resynch operation looks up all the valid APM policies for the selected nodes and redistributes all the APM policies.

**Modifying an Existing Policy**

To modify an existing policy, select the policy you want to modify on the Agent Policy Manager Configuration main menu and click **Modify**.

After a policy has been defined or distributed, you can modify the policy in two ways:

- Change **who** by adding or deleting target SmartSets from the policy.
When you use the SmartSet Editor to change the SmartSets used by an APM policy, the change to the SmartSet is distributed to the nodes in the SmartSet automatically. You do not need to redistribute the policy to the nodes in the SmartSet to implement the changes to the SmartSet.

- **Change what** by modifying the fields in the policy dialog box. See “Completing the APM Policy Dialog Boxes” on page 235 for additional information about what to put in the dialog box fields.

After you modify a policy, you must **distribute** the modified policy to the target nodes. You can choose to **distribute** the modified policy, **undo** the changes to the policy, or **delete** the policy completely.

After you modify a policy, the policy’s status is PendingModifyDistribute until it is distributed. When you distribute the modified policy, the changes are analyzed. Only the changes to the existing policy are distributed to each node that is impacted by the policy change.

### Viewing Pending Policies

If you have distributed or partially distributed policies and then modify the policy, you can view the modified policy before redistributing it. The modified policy has a status of PendingModifyDistribute. Select the **View Pending** button to see the modified policy. The **View Current** button shows the currently distributed version of the policy.

### Deleting an Existing APM Policy

To delete an APM policy, select the policy you want to remove, and then click **Delete** on the APM Configuration main menu. The status of the policy is changed to PendingDeleteDistribute. If you change your mind, click **Undo** to return the policy to its previous status. To distribute the delete operation to the nodes in the SmartSet, click **Distribute**.

On the Distribute dialog that is displayed, click **Start** to delete the policy on all nodes. This action removes the policy from the nodes. Messages are displayed to indicate the success or failure of the operation.

If any or all of the nodes cannot be reached, the policy has a status of PartiallyDeleted. If the problem was a timeout, such as if an agent was not available, you can redistribute manually by clicking **Distribute** again until all nodes are deleted. However, APM continues to periodically redistribute the delete operation as well. The policy remains in the database until APM is able to delete the policy from all nodes in the SmartSet. APM automatically deletes the policy from the database when the last node responds to the delete operation.

If a policy is in the PartiallyDeleted state, you can force deletion of the policy from the database by selecting it and clicking **Delete** again; however, this action can leave the policy still running in some of the nodes to which it has been distributed successfully and not deleted successfully. It might then be necessary to manually delete the policy from the hosts affected using the Tivoli NetView MLM Configuration interface on those Tivoli NetView MLM hosts.

Forcing a deletion from the database should only be done in cases where the hosts or agents that have failed deletion requests are removed from the network and will not return to receive another deletion request.
To keep a policy in the database, but remove the policy from all nodes in the network, follow this procedure:

1. Select the policy on the Agent Policy Manager Configuration main menu.
2. Click **Modify**. The Modify Policy dialog box is displayed.
3. Select all the SmartSet names under SmartSet Assignments.
4. Click **Unassign** and click on **Apply**.
5. On the Agent Policy Manager Configuration main menu, the policy now has a status of PendingModifyDistribute. Select the policy and click on **Distribute**.

   The status of the policy changes to Distributed.

To delete individual nodes that have failed deletion from the distribution list, follow this procedure:

1. Select the policy on the Agent Policy Manager Configuration main menu.
2. Select **Node Status** The Node Distribution Status dialog box is displayed.
3. Select the nodes to which you no longer want to retry the deletion request, and then click **Delete**. The nodes are deleted from the distribution list.

**Undoing a Change**

If you modify or delete a policy and then change your mind before you distribute it, you can undo the change by clicking **Undo** on the Agent Policy Manager Configuration main menu.

**Fixing an APM Policy**

If you have created an APM policy that prevents you from starting the APM configuration application or is causing performance problems, you can fix the policy in two ways:

- Use the **C5Maint** command to fix the fields in the policy definition.
- Use the **APMSetStatus** command to change policy status.

**Using the C5Maint Command**

To fix a problem APM policy using the **C5Maint** command, follow this procedure:

1. Stop the C5d daemon using the following command:
   ```bash
   /usr/OV/bin/ovstop C5d
   ```
2. Enter the following command:
   ```bash
   /usr/OV/bin/C5Maint -d
   ```

   The `/usr/OV/conf/C5.conf` file is created with all your APM policy definitions in it.

   The C5.conf file has the following format:

   ```
   operation:objID:list_flag:field_type_indicator:field_length\field_name:field_value_length,field_value
   ```

   **Note:** Do not change the objID, list_flag, field_type_indicator, field_length, or field_name fields. You can delete entire lines to remove node lists or variable lists, and you can change the field_value field and the field_value_length field to correspond with the changed value.

   Here is an example of the C5.conf file:
3. Edit the C5.conf file and delete all the policies except the one you want to change. The policy definition starts with Selection Name.

4. Change the field values you think are incorrect. Be sure to change the field length to correspond with the new field value.

5. Load the file into the APM database using the following command:
   ```
   C5Maint -l
   ```

   See the `C5Maint` man page for more information.

### Using the APMSetStatus Command

To change the policy status, follow this procedure:

1. Stop the C5d daemon using the following command:
   ```
   /usr/0V/bin/ovstop C5d
   ```

2. Enter the following command:
   ```
   APMSetStatus policy_name newStatus
   ```

   Replace `policy_name` with the name of the policy whose status you are changing. Replace `newStatus` with one of the following values:

   - **Distributed**
     Changing to this status causes the failed nodes list to be deleted from the policy.

   - **NeverDistributed**
     Changing to this status causes the failed nodes and distributed nodes lists to be deleted from the policy.

   - **PartiallyDistributed**
     Changing to this status does not affect the node lists.

   - **PartiallyDeleted**
     Changing to this status does not affect the node lists.
PendingDeleteDistribute
Changing to this status does not affect the node lists.

PendingModifyDistribute
Changing to this status requires that a modified version of the policy already exists in the database. You can confirm this by using the `ovobjprint mod_definitionName` command. Changing to this status causes failed and distributed nodes lists to be deleted from the policy.

Deleted
Changing to this status deletes the policy and any modified versions of the policy from the database, regardless of distribution status or existing policies in MLMs and SIAs. No distribution of the delete is done. Not changing the status to any status except PendingModifyDistribute causes any modified versions of the definition existing in the database to be deleted.

Note that the `APMSetStatus` command will not cause the executing environment to be updated in accordance with your policy changes. That is, all the agents and MLMs which are running versions of the definitions you might be changing will not be updated. This is a database update only. You must make sure that the MLMs and SIAs reflect the changes being made.

See the `APMSetStatus` man page for more information.

Copying an APM Policy from One System to Another

To copy an APM policy from one system to another, follow these steps:
1. Stop the C5d daemon using the following command:
   `/usr/OV/bin/ovstop C5d`
2. Enter the following command:
   `/usr/OV/bin/C5Maint -d`
   The `/usr/OV/conf/C5.conf` file is created with all your APM policy definitions in it. See “The C5.conf File” on page 233 for information about the format of the C5.conf file.
3. Edit the C5.conf file and delete all the system-specific information, such as failed and distributed node lists and invalid SmartSets. Change the distribution status to NeverDistributed. That is, change the field_value field corresponding to the dStatus field to NeverDistributed and change the field_value_length field to 16.
4. Load the file into the APM database on the system to which you want to copy the policy using the following command:
   `C5Maint -1`
   See the `C5Maint` man page for more information.

Completing the APM Policy Dialog Boxes

This section explains how to fill in the fields on the APM policy dialog boxes.
If you are familiar with Tivoli NetView MLM, the dialogs will be familiar to you. They are very similar to the dialog boxes in the Tivoli NetView MLM Configuration Application, with the added ability of assigning SmartSets to the policies. APM policy names and SmartSet names must be compliant with Tivoli NetView MLM requirements.

Threshold and Data Collection Policy Dialog Box

To define a threshold policy, select Threshold/Data Collection using the Policy Type button on the Agent Policy Manager Configuration main menu.

The following dialog box is displayed as shown in Figure 43.

![Threshold and Data Collection Dialog Box](image)

*Figure 43. Threshold and Data Collection Dialog Box*
The fields in this dialog box have the following meanings:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Values</th>
<th>Purpose and Description</th>
</tr>
</thead>
</table>
| **Name**   | Any unique character string. | **Purpose:** Specifies the name of the particular threshold entry.  
**Description:** The value in the Name field is used as a label and an instance ID, identifying each field for this threshold, and is appended to the corresponding object ID. The instance IDs for each field in the same entry are the ASCII values for each letter of the name. For example, the value `APP_TIME` causes the instance ID for this entry to be `65.80.80.95.84.73.77.69`, where each letter is an ASCII integer representation of `APP_TIME`. |
| **State**  | Valid values are: disabled, enabledThresholdOnly, enabledStoreOnly, enabledThresholdStore | **Purpose:** Determines whether the threshold entry is active and the thresholding actions that are being performed for this entry.  
**Description:** Their values and their meanings are described in the following list:  
- If the value is `disabled`, the entry is valid, but the MLM will not check the value of the specified MIB variable against this threshold.  
- If the value is `enabledThresholdOnly`, the MLM is checking the retrieved MIB variable values against this threshold, then discarding the data.  
- If the value is `enabledStoreOnly`, the MLM is storing the retrieved MIB variable values in a file on the MLM host during the polling process; the values are not checked against the threshold.  
- If the value is `enabledThresholdStore`, the MLM is checking the retrieved MIB variable values against this threshold and storing the values in a file on the MLM host. |
| **Description** | Any valid character string. | **Purpose:** The general purpose of the threshold being performed.  
**Description:** Completing this field is not mandatory, but it is recommended. Describe the purpose and function of the threshold. |
| **Poll Time** | Valid values include an integer followed by one of these letters: d = days, h = hours, m = minutes, s = seconds | **Purpose:** Indicates the time period that should elapse before the next threshold polling operation is invoked.  
If the letter is omitted, the default is minutes. For example, the value `1h10m` means to start this threshold polling operation 1 hour and 10 minutes following the last polling operation. |
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Values</th>
<th>Purpose and Description</th>
</tr>
</thead>
</table>
| **MIB Object** | Any MIB variable in numeric, dot-notation format, followed by an instance ID in numeric or nonnumeric format. For multiple instances, a wildcard is valid. | **Purpose:** Indicates the MIB variables on which thresholding is to be performed. Do not begin a MIB object with an alias or node address. Because you are assigning the threshold to a collection, it is not necessary to specify a remote node.  
**Description:** You can specify the MIB variable in the following ways:  
- Specify one unique instance to be checked  
- Specify all instances of a variable  
  
For example, to perform threshold checking for the specific instance interface 1 of the following MIB variable:  
mib-2.interfaces.ifTable.ifEntry.ifInErrors  
the value would be: .1.3.6.1.2.1.2.2.1.14.1.  
To perform threshold checking for all instances of the previous MIB variable, the value is .1.3.6.1.2.1.2.2.1.14.*.  
You can use the alphanumeric string to indicate the instance to be checked. For example, to perform threshold checking for the MIB variable sm6kSystemFileSystemPercentUsed for the /usr directory instance, the value is .1.3.6.1.4.1.2.6.12.2.5.2.1.4./usr.0.  
By default, thresholds for Counter type variables are computed by calculating the change per second in the sampled values and checking these delta values against the threshold or rearm value. For Gauge and Integer type variables, the actual variable value is checked against the threshold or rearm value. To override this default behavior, add the keyword *delta* or *value* in front of the condition, separating the keyword and condition by a space. |
| **Threshold Condition and ReArm Condition** | Valid condition values are:  
- =  
- <  
- <=  
- >  
- >=  
- &  
- |  
- changes  
- doesNotChange  
- exists  
- doesNotExist  
- !<condition> | **Purpose:** Indicates the condition used when checking retrieved MIB values against the threshold or rearm value.  
**Description:**  
- =, <, <=, >, and >= mean the retrieved value being checked against the threshold or rearm value must be either equal to, less than (or equal to), greater than (or equal to) the value, respectively.  
- Add an ! to the beginning of a condition to indicate what the value must not be.  
- & and | cause the retrieved values and threshold or rearm value to be combined with a logical AND or OR statement, respectively. A non-zero (0) result meets the condition; the trap or command operation is performed.  
- changes and doesNotChange cause watchdog operations to be performed. If a retrieved value changes or does not change, respectively, between consecutive polls, the condition is met. Threshold or rearm values are not used.  
- exists and doesNotExist check if the MIB variable exists. To check for an instance of the Tivoli NetView subagent (trapgend daemon) within the MLM Process table, use exists and .1.2.6.1.4.1.2.6.12.2.7.2.1.2.trapgend.* as the variable.  
By default, thresholds for Counter type variables are computed by calculating the change per second in the sampled values and checking these delta values against the threshold or rearm value. For Gauge and Integer type variables, the actual variable value is checked against the threshold or rearm value. To override this default behavior, add the keyword *delta* or *value* in front of the condition, separating the keyword and condition by a space. |
### Table 21. Threshold Policy Dialog Box Fields (continued)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Values</th>
<th>Purpose and Description</th>
</tr>
</thead>
</table>
| Value and Rearm Value | Depending on the MIB variable, the value can be a number, a floating point number, or a character string. | **Purpose:** Specifies the threshold or rearm value for a MIB variable against which retrieved values are checked.  
**Description:** This value must match the type of MIB variable that is being checked. If the MIB variable is numeric, use a number and a numeric check is done. If the MIB variable is a display string, and both the retrieved variable and this value can be converted to a floating point number, a floating point check is done. Otherwise, a string check is done. For example, if the MIB variable that is checked is a Counter type, the value should be an unsigned integer. |
| Specific or ReArm Specific | Any valid SNMP enterprise-specific trap number. | **Purpose:** This field is on the Threshold Actions and Rearm Actions dialog boxes. It specifies the SNMP enterprise-specific trap which is to be sent when the threshold or rearm condition is met. Use the value 1 for arm and the value 2 for rearm. If this value is 0 (zero), no trap is sent. |
| Enterprise or ReArm Enterprise | The valid SNMP enterprise ID, in dot notation, for the specified trap. | **Purpose:** This field is on the Threshold Actions and Rearm Actions dialog boxes. It specifies the SNMP enterprise ID for the enterprise-specific trap which is to be sent when the threshold or rearm condition is met. Use the .1.3.6.1.4.1.2.6.12.5.1 for the enterprise ID. If this field is blank, the Threshold Table enterprise ID is used. |
| Trap Description or ReArm Trap Description | Any valid character string. The description can use a set of defined environment variables. | **Purpose:** This field is on the Threshold Actions and Rearm Actions dialog boxes. It indicates the general reason for sending the trap. It is the first variable in the trap that is sent to the operator. |
| Command To Execute or ReArm Command To Execute | Any valid command can be specified. The command can use a set of defined environment variables. | **Purpose:** This field is on the Threshold Actions and Rearm Actions dialog boxes. It specifies a command to be executed by the MLM when the threshold or rearm condition is met.  
**Description:** The command is executed using the environment in which the MLM is running. If this field is blank, no command will be executed.  
See "Matched, Armed, and Disarmed Command Environment Variables" on page 275 for information about environment variables that can be used in scripts. |

---

### File Monitor Policy Dialog Box

To define a file monitor policy select **File Monitor** using the Policy Type button on the Agent Policy Manager Configuration main menu. The following dialog box is displayed as shown in [Figure 44 on page 240](#).
The fields in this dialog box have the following meanings:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Values</th>
<th>Purpose and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Any unique character string that is a valid Tivoli NetView MLM name. No spaces are permitted in the string.</td>
<td>Purpose: Specifies the name to be used as a label and instance ID identifying each field for this command entry. The name is appended to the corresponding object ID. The instance IDs for each field in the same entry are the ASCII values for each letter of the name. For example, a name of RCMON causes the instance ID for this entry to be 82.67.77.79.78, which is an ASCII representation of the name RCMON.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Values</td>
<td>Purpose and Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>-------------------------</td>
</tr>
<tr>
<td><strong>State</strong></td>
<td>Valid values are:</td>
<td><strong>Purpose</strong>: Specifies the current state of the file monitor table.</td>
</tr>
<tr>
<td></td>
<td>enabledFromBegin</td>
<td><strong>Description</strong>: The values mean:</td>
</tr>
<tr>
<td></td>
<td>disabled</td>
<td>• If the value is <strong>disabled</strong>, file monitoring is turned off.</td>
</tr>
<tr>
<td></td>
<td>enabled</td>
<td>• If the value is <strong>enabled</strong>, file monitoring is turned on. Actual monitoring starts when the activation time you specified is reached. Tivoli NetView MLM begins searching at the end of the file, unless it determines that the file has wrapped or has decreased in size; if so, Tivoli NetView MLM starts searching at the top of the file. In this way, Tivoli NetView MLM uses only the newest file information. When the string you specified is found in the file, a trap is issued, and any action you specified in the Command to Execute field is run.</td>
</tr>
<tr>
<td></td>
<td>enabledFromBegin</td>
<td>• If the value is <strong>enabledFromBegin</strong>, file monitoring is active. Monitoring starts at the top of the file. You might specify this value if the file to be monitored is actually being generated each polling interval by a command that you specified to be run before each search, or if you always want the file to be searched from top to bottom each polling interval. Note that this state requires more CPU processing than the <strong>enabled</strong> state.</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Any valid character string.</td>
<td><strong>Purpose</strong>: States the general purpose of the file being monitored.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Description</strong>: Describes the purpose of the monitored file and explains the action that is taken when a search string or some other change to the file is detected.</td>
</tr>
<tr>
<td><strong>Poll Time</strong></td>
<td>Valid values include an integer followed by one of these letters:</td>
<td><strong>Purpose</strong>: Specifies the poll interval.</td>
</tr>
<tr>
<td></td>
<td>d = days</td>
<td><strong>Description</strong>: This variable specifies how often SIA should check the monitored file for the specified string.</td>
</tr>
<tr>
<td></td>
<td>h = hours</td>
<td>If the letter is omitted, the default is minutes. Multiple units can be specified in this field. For example, you can specify a polling interval of 1h45m (1 hour and 45 minutes). If you do not specify a polling interval, the default is 10 seconds.</td>
</tr>
<tr>
<td></td>
<td>m = minutes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>s = seconds</td>
<td></td>
</tr>
<tr>
<td><strong>File to Monitor</strong></td>
<td>Any valid character string.</td>
<td><strong>Purpose</strong>: Specifies the path name where the monitored file resides.</td>
</tr>
<tr>
<td>(Full Path Name)</td>
<td></td>
<td><strong>Description</strong>: Specify the name of the file to be monitored, including the fully qualified path where the file resides. The file must be stored on the same node as the SIA, and a root user must have read permission for the file. Monitoring of the file differs, depending on whether or not the file exists.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• If the file already exists, monitoring begins with the last byte position in the file (the most recent records).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• If the file does not exist, when you enable file monitoring, Tivoli NetView MLM waits until the file is created and begins searching for the string at the beginning of the file.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Values</td>
<td>Purpose and Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Monitor Type</td>
<td>Values are:</td>
<td><strong>Purpose:</strong> Specify the type of monitoring to be performed.</td>
</tr>
<tr>
<td></td>
<td>• string</td>
<td><strong>Description:</strong> This field specifies the type of monitoring to be performed on the</td>
</tr>
<tr>
<td></td>
<td>• dataChange</td>
<td>specified string or condition.</td>
</tr>
<tr>
<td></td>
<td>• statusChange</td>
<td>• If the value is <strong>string</strong>, the SIA watches the specified file for the string</td>
</tr>
<tr>
<td></td>
<td>• strDataStatus</td>
<td>specified in the String to Monitor field. When the string appears, the specified</td>
</tr>
<tr>
<td></td>
<td>• notExist</td>
<td>actions are done.</td>
</tr>
<tr>
<td></td>
<td>• exist</td>
<td>• If the value is <strong>dataChange</strong>, the SIA watches the file for any change to the</td>
</tr>
<tr>
<td></td>
<td>• all</td>
<td>contents of the file, such as added or deleted characters. If any data change occurs,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the specified actions are done.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• If the value is <strong>statusChange</strong>, the SIA watches the file for any change to the file</td>
</tr>
<tr>
<td></td>
<td></td>
<td>status, including the file owner, file group, and file permissions. The file mode,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>user ID, and group ID that will be used for comparison are shown in the File Mode, User</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ID, and Group ID fields on the File Monitor window. The actual file mode, user ID, and</td>
</tr>
<tr>
<td></td>
<td></td>
<td>group ID can be changed by modifying these fields in the File Monitor window. If the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>file mode, file owner, or file group changes, the specified actions are done.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• If the value is <strong>strDataStatus</strong>, the SIA watches for any of the previous three</td>
</tr>
<tr>
<td></td>
<td></td>
<td>types of changes (data changes, the appearance of a string, or file status changes).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If any of these changes occur, the specified actions are done.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• If the value is <strong>notExist</strong>, the SIA does the specified actions if the monitored</td>
</tr>
<tr>
<td></td>
<td></td>
<td>file disappears (such as if it is erased).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• If the value is <strong>exist</strong>, the SIA does the specified actions if the monitored file</td>
</tr>
<tr>
<td></td>
<td></td>
<td>appears.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• If the value is <strong>all</strong>, the SIA performs all types of monitoring (basically any</td>
</tr>
<tr>
<td></td>
<td></td>
<td>change at all to the file).</td>
</tr>
<tr>
<td>Traps</td>
<td>Valid values are:</td>
<td><strong>Purpose:</strong> Specifies the types of traps that you want to have forwarded by the file</td>
</tr>
<tr>
<td></td>
<td>• send</td>
<td>monitoring function.</td>
</tr>
<tr>
<td></td>
<td>• noSend</td>
<td><strong>Description:</strong> The values mean:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• If the value is <strong>send</strong>, the file monitoring function sends all types of traps. The</td>
</tr>
<tr>
<td></td>
<td></td>
<td>trap destination is specified in the snmp.conf file.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• If the value is <strong>noSend</strong>, traps are not sent.</td>
</tr>
<tr>
<td>String to Monitor</td>
<td>Any valid character string.</td>
<td><strong>Purpose:</strong> Specifies the string that will be monitored.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Description:</strong> Specifies the file string that SIA should search for in the file.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The string or pattern can be any limited regular expression (RE) in the style of the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UNIX ed or egrep commands.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note that if you use anchor symbols (^ or $) in the expression, the string is only</td>
</tr>
<tr>
<td></td>
<td></td>
<td>found if the line containing the anchor symbols is terminated with a new line (\n)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>character.</td>
</tr>
<tr>
<td>Case Sensitive</td>
<td>Valid values are:</td>
<td><strong>Purpose:</strong> Specifies whether the search is case sensitive.</td>
</tr>
<tr>
<td></td>
<td>• case – Perform a case-sensitive search</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• ignoreCase – Ignore case on search</td>
<td></td>
</tr>
<tr>
<td>Field Name</td>
<td>Values</td>
<td>Purpose and Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Activation Schedule</td>
<td>Any valid character string for activation time</td>
<td><strong>Activation</strong>: Use this field to delay the onset of file monitoring. Specify time in the format \texttt{HH:MM}, where \texttt{HH} is an integer in the range 0–23 and \texttt{MM} is an integer in the range 0–59. If this field is not set, the time is set to 00:00. Select the days of the week when you want file monitoring to be active. Deactivation Time: Use this field to set the time when file monitoring will be deactivated. Specify time in the format \texttt{HH:MM}, where \texttt{HH} is an integer in the range 0–23 and \texttt{MM} is an integer in the range 0–59. If this field is not set, the time is set to 00:00. Select the days of the week when you want file monitoring to be deactivated.</td>
</tr>
<tr>
<td>Automated Actions.</td>
<td>Any valid character string</td>
<td><strong>Purpose</strong>: The command to be run before monitoring begins, and the command to run if the string is found. <strong>Description</strong>: Click this button to set up commands to be run before file monitoring and during monitoring when a condition is found. Command to Execute Before Monitor: Indicates the command to be run before searches for the specified string or other change to the file are performed. This MIB variable can be used to generate, modify, or translate the file prior to the search. Command to Execute if Monitor Type Condition is Met: Indicates the command to be run automatically when a certain string of text or some other kind of change occurs in the monitored file. For example, you could run a command to expand the size of a file system if you are monitoring for a message that the file system is full. Indicates the command to be run before searches for the specified string or other change to the file are performed.</td>
</tr>
<tr>
<td>SmartSet Assignments</td>
<td></td>
<td>Use this field to assign SmartSets to this policy. Click <strong>Assign</strong> to open a dialog box for selecting SmartSets.</td>
</tr>
</tbody>
</table>

**Command Policy Dialog Box**

To define a command policy, select **Command** using the Policy Type button on the Agent Policy Manager **Configuration** main menu. The following dialog box is displayed as shown in **Figure 45 on page 244**.
Figure 45. Command Policy Dialog Box

The fields in this dialog box have the following meanings:
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Values</th>
<th>Purpose and Description</th>
</tr>
</thead>
</table>
| Name       | Any unique ASCII character string that is a valid Tivoli NetView MLM name. Spaces are not valid in the character string. | **Purpose:** Specifies the name of a particular command in the table.  
**Description:** The value in the Name field is used as a label and an instance ID, identifying each field for this command, and is appended to the corresponding object ID. The instance ID for each field in the same entry is the ASCII values for each letter of the name. For example, the value `APP_TIME` causes the instance ID for this entry to be `65.80.80.95.84.73.77.69`, where each letter is an ASCII integer representation of `APP_TIME`. Use short names if possible. |
| State      | Valid values are: enabled disabled invalid | **Purpose:** Identifies the current availability of a command.  
**Description:** The values mean:  
- If the value is **enabled**, the command is available for execution.  
- If the value is **disabled**, the command is valid, but it is not currently available for execution.  

The default value is **disabled**.  
If you click **Delete** on the APM Configuration main menu, the State field is changed to **invalid**, which causes the selected entry to be deleted. |
| Description| Any valid text string. | **Purpose:** States the general purpose of the command.  
**Description:** The description character string can be as detailed as necessary. This field is for informational purposes. This field is not mandatory, but is recommended. |
| Get Command|  | **Purpose:** Specifies the command that is to be executed in the Korn shell, the kernel memory of the operating system, or the shared memory segment of a user application for an SNMP get request.  
**Description:** An SNMP get request must be issued to the defined Output Result Index (set in the Result Type field). This index is used for conversion to the output types display string, integer, counter, and gauge. The results are returned to the MIB variable that is specified by the defined Output Result Index (set in the Result Type field) and propagated to other result types. Therefore, if the index is set to **displaystring**, an SNMP get request to the DisplaystringResult causes the get command expression to execute and return a displaystring. Commands use preset environment variables in the command string.  

To use shared or kernel memory, specify the keyword **SHARE_MEMORY** or **KERNEL_MEMORY** as the first 14 characters of the command.  

For more information about using the Get Command field, refer to the **Tivoli NetView MLM User’s Guide**. |
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Values</th>
<th>Purpose and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set Command</td>
<td></td>
<td><strong>Purpose:</strong> Specifies the command to be executed on an SNMP set request or a set request to a shared memory segment. <strong>Description:</strong> This is a way to specify a value for a command that is to be run in the Korn shell. The value of the set is placed in the environment variable SM6K_COMMAND_SET_VALUE, which can be used on the command. An SNMP set request must be issued to the defined Output Result Index (set in the Result Type field). This index is used for conversion to the output types displaystring, integer, counter, and gauge. Therefore, if the index is set to displaystring, an SNMP set request to the DisplaystringResult (the Result field) causes the Set command expression to execute using the SM6K_COMMAND_SET_VALUE. The output of the command is placed in the appropriate MIB variable defined by the Output Result Index (set in the Result Type field). Commands can use preset environment variables in the command string. If the command execution results in a nonzero return code, the set request fails and a BAD VALUE message is returned. To use a set request in shared memory, specify the keyword SHARED_MEMORY: as the first 14 characters of the command. This notifies the SIA that a set shared memory operation is requested. For more information about using the Set Command field, refer to the Tivoli NetView MLM User’s Guide.</td>
</tr>
<tr>
<td>Time Out</td>
<td>Any valid integer, from 0 to 4294967295, representing a time period.</td>
<td><strong>Purpose:</strong> The length of time, in seconds, for the get or set command, or memory operation to execute. <strong>Description:</strong> The time out value is critical. It allows the SIA to run the command, or it allows the command and memory operation to be cancelled if the command does not complete. The default value is 3 seconds. If you specify the value 0, the command or memory operation is executed with no time out; no output or result is returned.</td>
</tr>
<tr>
<td>Time To Live</td>
<td>Any valid integer, from 0 to 4294967295, representing a time period.</td>
<td><strong>Purpose:</strong> Specifies the length of time, in seconds, before the command is executed again in response to an SNMP query. <strong>Description:</strong> Shows the amount of time in seconds that must pass before another command is executed in response to an SNMP get request. If another get request is issued and the time to live interval has not passed, the command specified on the get request is not executed. Use this variable to allow time for multiple get requests on the output from a previous command.</td>
</tr>
</tbody>
</table>
Table 23. Command Policy Dialog Box Fields (continued)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Values</th>
<th>Purpose and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result Type</td>
<td>Valid values are: displaystring integer counter gauge</td>
<td><strong>Purpose:</strong> Indicates the type of output request, or Output Result Index, to be used for operation results. <strong>Description:</strong> To execute the defined SNMP get or set command expression, you must issue the get or set request to an Output Result Index. Based on the Result Type value, the results for the command are saved to a predefined corresponding MIB variable. If the result is an integer, it can be propagated to the other predefined MIB variables. For example, if the Result Type value is integer, when the command is run, the command output is converted to an integer, saved in the IntegerResult MIB variable, and propagated to all other variables based on IntegerResult.</td>
</tr>
<tr>
<td>Row Index and Column Index</td>
<td>Any valid integer in the range of 0 to 4294967295.</td>
<td><strong>Purpose:</strong> Allows command output parsing for individual rows and columns. <strong>Description:</strong> Parsing for rows is based on new lines; parsing for columns is based on white spaces. Use these values together to control the output that is returned. A Row Index value of 0 (zero) returns all rows. A Column Index value of 0 (zero) returns all columns. Used together, the following results are possible: ▪ If both the Row and Column Index values are 0 (zero), all output is returned. ▪ If the Row Index value is not 0 and the Column Index value is 0 (zero), the specified row is returned. ▪ If the Row Index value is 0 (zero) and the Column Index value is not 0 (zero), the specified column is returned. ▪ If neither the Row Index value nor Column Index value are 0 (zero), only the row and column intersection field is returned. For example, if the Row Index value is 0 and the Column Index value is 0, the <code>ps -ef</code> command could return the following results: cag 19765 1 0 2:55:25 - 0:00 xterm ja 29752 43830 0 1:24:51 pts/25 0:00 bm/ksh In this example, if the Row Index value was 2 and the Column Index was 2, the output from running the <code>ps -ef</code> command would be 29752.</td>
</tr>
</tbody>
</table>

**Filter Policy Dialog Box**

To define a filter policy, select **Filter** using the Policy Type button on the Agent Policy Manager Configuration main menu. The following dialog box is displayed as shown in Figure 46 on page 248.
The fields in this dialog box have the following meanings:

Figure 46. Filter Policy Dialog Box
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Values</th>
<th>Purpose and Description</th>
</tr>
</thead>
</table>
| Name       | Any unique character string. | **Purpose:** Specifies the name of the particular filter rule.  
**Description:** A value in the Name field is used as a label and an instance ID, identifying each filter rule field, and is appended to the corresponding object ID. The instance IDs for the fields in the same filter rule are the ASCII values for each letter of the name. For example, for the filter rule name *Test*, the instance ID is 84.101.115.116. |
| State      | Valid values are: Disabled, Enabled, Invalid | **Purpose:** Specifies the current availability of the filter rule.  
**Description:**  
- **Disabled** means that the filter rule is dormant and is not used when traps are received.  
- **Enabled** means that the filter rule can be used in the filter evaluation process when a trap is received. Whether or not the filter rule will actually be used at a specific time is determined by the values in the Activation/Deactivation fields. See the Activation Schedule field to determine whether a rule is currently being used.  
When you click **Delete** the Agent Policy Manager Configuration main menu, the State field is changed to **invalid**, which causes the selected entry to be deleted. |
| Description| Any valid character string. | **Purpose:** Specifies the general purpose of the filter rule.  
**Description:** This field is not mandatory, but is recommended. This field should describe the purpose and actions of the filter. |
| Enterprise | Valid values are:  
- Enterprise ID in dot notation (fully qualified)  
- Enterprise ID in dot notation (with wildcard value)  
- Alias name (from the Alias Table) | **Purpose:** The enterprise ID is compared to the enterprise ID in the received trap. The IDs must match.  
**Description:** You can specify a single fully-qualified enterprise ID in this field, or you can indicate multiple enterprises in one expression by using a wildcard to specify a partial enterprise ID. The asterisk must be the last character in the string. You can also indicate an alias name, defined in the Alias Table, that represents an enterprise ID. |
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Values</th>
<th>Purpose and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent Address</td>
<td>Valid values are:</td>
<td><strong>Purpose:</strong> Specifies the set of agents, or nodes, that might have generated the received trap.</td>
</tr>
<tr>
<td></td>
<td>• Alias name (from the Alias Table)</td>
<td><strong>Description:</strong> The Agent Address expression can be a single address value, or a list of multiple address values separated by commas.</td>
</tr>
<tr>
<td></td>
<td>• Host name</td>
<td>An IP address expression should be in the format of 4 dot-separated address subexpressions. Each subexpression can be one of the following:</td>
</tr>
<tr>
<td></td>
<td>• IP address expression</td>
<td>• A decimal number in the range 0–255.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• An inclusive range of 0–255 separated by a dash (for example, 0–127 or [0–127]).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• An asterisk.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Each address subexpression represents 1-byte of the agent address for which the range of values 0–255. An asterisk represents any byte in the range of 0–255, which can be interpreted as [0–255].</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For example, a filter rule with the IP address expression 9.[67-69].5.* matches the received-trap agent address 9.68.5.181.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you leave this field blank for a filter rule, all agent addresses in received traps will be considered a match.</td>
</tr>
<tr>
<td>Generic Expression</td>
<td>Valid values are:</td>
<td><strong>Purpose:</strong> Specifies the generic trap numbers that will be considered a match for this filter rule.</td>
</tr>
<tr>
<td></td>
<td>0 (Cold Start)</td>
<td><strong>Description:</strong> You can enter a single or multiple generic trap numbers separated by commas. This field does not support pattern matching characters (wildcards) or number ranges.</td>
</tr>
<tr>
<td></td>
<td>1 (Warm Start)</td>
<td>If you specify an enterprise-specific trap in the Specific Expression field, 6 is required in this field.</td>
</tr>
<tr>
<td></td>
<td>2 (Link Down)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 (Link Up)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 (Authentication Failure)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5 (EGP Neighbor Loss)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6 (Enterprise Specific)</td>
<td></td>
</tr>
</tbody>
</table>
**Table 24. Filter Policy Dialog Box Fields (continued)**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Values</th>
<th>Purpose and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific Expression</td>
<td>Any valid enterprise-specific trap number.</td>
<td><strong>Purpose:</strong> Specifies the enterprise-specific trap numbers that will be considered a match for this filter rule.</td>
</tr>
</tbody>
</table>
|                 |                                                   | **Description:** You can enter a single specific trap number or multiple specific trap numbers separated by commas. You can specify a range of specific trap numbers separated by a dash, for example, 2–5. Also, you can associate an enterprise ID with a group or range of enterprise-specific trap numbers. This association enables you to pair enterprise-specific traps from one vendor with those of another vendor in the same filter rule. To associate an enterprise ID with a group of trap numbers, enclose the trap numbers in parentheses and brackets, as shown in the following example:
|                 |                                                   | .1.3.6.1.4.1.2.6.12([21-23]), .1.3.6.1.4.1.2.6.12(21, 23), 25 |
|                 |                                                   | If there is an enterprise ID associated with the specific-trap number, the number is matched with that enterprise ID. If there is no enterprise ID associated with the specific-trap number, the number is matched with the default enterprise ID. |
|                 |                                                   | This field is checked only if the number 6 is entered in the Generic Expression field. If this field is left blank and the number 6 is specified in the Generic Expression field, all enterprise-specific trap numbers contained in received traps are considered matches. |

<table>
<thead>
<tr>
<th>Variable Expression</th>
<th>Any valid MIB expression</th>
<th><strong>Purpose:</strong> Specifies the MIB variable expressions in the traps that will be considered a match for this filter rule.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>Description:</strong> MIB variables are shown in an expression that is set off with relational operators (&lt;, &gt;, &lt;=, &gt;=, ==, !=) and logical operators (&amp;&amp;,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Here is an example of a valid MIB variable expression: <em>( $SM6K_TRAP.ENTERPRISE == .1.3.6.1.4.1.2.6.12 ) &amp;&amp; ( $SM6K.TRAP.SPECIFIC_NUM &lt;= 10 )</em></td>
</tr>
<tr>
<td>Field Name</td>
<td>Values</td>
<td>Purpose and Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Action          | Valid values are:              | **Purpose:** Determines whether traps matching the filter rule are forwarded to the Tivoli NetView hosts that are specified in the Trap Destination field. If the Trap Destination field for this filter rule is blank, the trap destination is obtained from the Trap Destination Table.  
**Description:**  
- **SendTraps** indicates that matching traps will be forwarded to the specified top-level managers.  
- **BlockTraps** indicates that matching traps are not forwarded. However, the MLM will forward the trap if it matches another filter rule that specifies **SendTraps**.  
- **ThrottleTraps** indicates that the MLM might or might not send the matching trap, depending on the configured throttle criteria. This criteria is specified in the Throttle Settings dialog box. |
| Throttle Type   | Valid values are:              | **Purpose:** This field is on the Throttle Settings dialog box. It specifies whether the throttle function will forward the first of N matching traps to the top-level manager or the traps received after N.  
**Description:**  
- **SendFirstN** means to forward the first N traps and block those traps received after N.  
- **SendAfterN** means to block the first N traps and forward those traps received after N.  
The value for N is defined in the Arm Count field. |
| Arm Count       | Any integer ≥1                 | **Purpose:** This field is on the Throttle Settings dialog box. It indicates the number of matching traps to receive before the throttle is armed.  
**Description:** Based on the selected Throttle Type, an armed throttle either allows the first N matching traps to be forwarded and blocks subsequent matching traps (SendFirstN), or blocks the first N matching traps and forwards subsequent matching traps (SendAfterN). |
| Armed Command   | Any valid command can be specified. The command invokes a set of defined environment variables. | **Purpose:** This field is on the Throttle Settings dialog box. It specifies a command to be executed by the MLM when the number of received matching traps matches the value for the Arm Count, and the throttle is armed.  
**Description:** The command is executed using the environment in which the MLM is running. For a descriptive list of valid environment variables that can be used by the command, see [“Matched, Armed, and Disarmed Command Environment Variables” on page 275](#). |
## Table 24. Filter Policy Dialog Box Fields (continued)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Values</th>
<th>Purpose and Description</th>
</tr>
</thead>
</table>
| Disarm Timer  | Valid values include an integer followed by one of these letters:       | **Purpose:** This field is on the Throttle Settings dialog box. It specifies the amount of time to pass, starting with the receipt of the first matching trap, before resetting or disarming the throttle. If the throttle is disarmed at the end of the time period, the **disarmed** command is executed.  
**Description:** If the letter is omitted, the default is minutes. For example, the value **1h10m** means to start this threshold polling operation 1 hour and 10 minutes following the last polling operation.  
The Disarm Timer and the Disarm Count methods for disarming a throttle can be used together.  
With a Disarm Count of greater than 0 (zero), it is possible for the throttle to be disarmed (by reaching the Disarm Count value) before the Disarm Timer period has elapsed.  
Using the Disarm Timer method, the timer is checked when a trap matching the filter rule is received. If the amount of time specified has elapsed since the throttle was started, the throttle is disarmed. If the amount of time specified has not elapsed since the throttle was started, the throttle action continues as defined.  
The timer is started when the throttle is started (the first matching trap is received), not when the throttle is armed (the Arm Count criteria is met and the **armed** command is executed). This means that it is possible for the throttle not to be armed before the time period has elapsed. Consequently, the time period elapses, but the throttle is only reset, not disarmed. Therefore, the **disarmed** command is not executed.                                                                 |
|               | d = days  
               | h = hours  
               | m = minutes  
               | s = seconds                                                                 |                                                                                                               |
| Disarm Count  | Any valid integer value.                                                | **Purpose:** This field is on the Throttle Settings dialog box. It indicates the number of matching traps that must be received to cause the throttle to be reset, or disarmed.  
**Description:** This is one method of disarming a throttle. Using this method, a count of matching traps is incremented when a trap matching the filter rule is received. When the number equals the defined Disarm Count, the throttle is disarmed. The counter starts over the next time the throttle is armed.  
This value is a delta, referring to the number of matching traps received after the throttle has been armed. This value is ignored if set to 0 (zero).  
Another method is to use the Disarm Count with the Disarm Timer. With a value set for Disarm Timer and a Disarm Count greater than 0 (zero), it is possible for the throttle to be disarmed (by the Disarm Timer period elapsing) before the Disarm Count value is reached.                                                                 |

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Values</th>
<th>Purpose and Description</th>
</tr>
</thead>
</table>
| Disarmed Command | Any valid command can be specified. The command can invoke a set of defined environment variables. | **Purpose:** This field is on the Throttle Settings dialog box. It specifies a command to be executed by the MLM when the currently armed throttle is disarmed. The throttle can be disarmed when the number of received matching traps matches the Disarm Count value, or when the specified Disarm Timer period has elapsed.  
**Description:** The command is executed using the environment in which the MLM is running. For a descriptive list of valid environment variables that can be used by the command, see “Matched, Armed, and Disarmed Command Environment Variables” on page 275. |
| Activation       | Any valid character string for activation time                         | **Purpose:** Specifies the time and days of the week when the filter policy is to be activated and deactivated.  
**Activation:** Use this field to delay the onset of filtering. Specify time in the format HH:MM, the hour (HH) can be an integer in the range of 0–23. The minute (MM) can be an integer 0–59. If this field is not set, the time is set to 00:00.  
Select the days of the week when you want filtering to be active.  
**Deactivation Time:** Use this field to set the time when filtering will be deactivated. Specify time in the format HH:MM, the hour (HH) can be an integer in the range of 0–23. The minute (MM) can be an integer 0–59. If this field is not set, the time is set to 00:00.  
Select the days of the week when you want filtering to be deactivated. |
| Trap Destinations| Host+service or host+protocol name/port, where host can be one of the following:  
- IP addresses  
- Alias names (from the Alias Table)  
- Host names | **Purpose:** This field is on the Automated Actions dialog box. It specifies the nodes to which a matching trap will be sent.  
**Description:** The values in this field will be used instead of the entries in the Trap Destination Table. If this field is left blank, the addresses defined in the Trap Destination Table are used. If duplicate addresses are specified for the destination of a trap, only one instance of the trap is sent to the address.  
You can use one of the following formats to specify the port number and protocol for trap delivery (UDP or TCP):  
- Host+port number/protocol name  
- Host+protocol name/port number  
- Host+service  
The service must be the name of a service listed in the /etc/services file that corresponds to a port number/protocol name.  
If you do not define the protocol name or port number, the default value is used. The default protocol name is udp: the default port number is 162. |
| Matched Command  | Any valid command can be specified. The command can use a set of defined environment variables. | **Purpose:** This field is on the Automated Actions dialog box. It specifies a command to be executed by the MLM when a received trap matches the filter rule.  
**Description:** The command is executed using the environment variables in which the MLM is running. For a descriptive list of valid environment variables, see “Matched, Armed, and Disarmed Command Environment Variables” on page 275. |
Trap Destination Policy Dialog Box

To define a trap destination policy, select Trap Destination using the Policy Type button on the Agent Policy Manager Configuration main menu. The following dialog box is displayed as shown in Figure 47.

![Figure 47. Trap Destination Policy Dialog Box](image)

The fields in this dialog box have the following meanings:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Values</th>
<th>Purpose and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Any unique character string.</td>
<td><strong>Purpose:</strong> Specifies the name of the particular trap destination entry.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Description:</strong> The value in the Name field is used as a label and an instance ID, identifying each field for this entry, and is appended to the corresponding object ID. The instance ID for the fields in the same entry are the ASCII values for each letter of the name. For example, for the entry name Test, the instance ID is 84.101.115.116.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Values</td>
<td>Purpose and Description</td>
</tr>
<tr>
<td>------------</td>
<td>------------------------------------------------------------------------</td>
<td>-------------------------</td>
</tr>
</tbody>
</table>
| State      | Valid values are: disabled, enabled, invalid                          | **Purpose:** Indicates the current state of the trap destination entry. **Description:**  
• Disabled means that the trap destination is not used as a destination for received traps.  
• Enabled means that the entry is used as a default destination for the traps that are received by the MLM.  
When you click **Delete**, the State field is changed to **invalid**, which causes the selected entry to be deleted. |
| Destination| Host+service or host+protocol name/portnumber, where host can be one of the following:  
• IP addresses  
• Alias names (from the Alias Table)  
• Host names | **Purpose:** Specifies the nodes to which the MLM will forward traps. **Description:** Multiple nodes can be specified in a list, with each name separated by a comma or white space.  
You can specify the port number to which traps will be delivered and the protocol to be used in trap delivery (UDP or TCP) by using one of the following formats:  
• Host+port number/protocol name  
• Host+protocol name/port number  
• Host+service  
The service must be the name of a service listed in the /etc/services file, which corresponds to a port number/protocol name.  
If you do not define the protocol name or port number, the default value is used. The default protocol is **udp**: the default port number is **162**. |
| Mask       | The mask value can be set using values in decimal, hex (0x##), or octet (0###) notation. | **Purpose:** Provides a filtering mechanism, preventing certain traps from being sent to the defined Destination. **Description:** The value is a numeric representation of a bit mask. Bits corresponding to the generic trap numbers 0–6 are used in the mask to indicate which traps will be forwarded to the defined nodes. The bits are read from left to right, with the leftmost bit representing the generic trap number 0, the next bit representing the generic trap number 1, and so on. The generic trap numbers are:  
• 0 = Cold Start  
• 1 = Warm Start  
• 2 = Link Down  
• 3 = Link Up  
• 4 = Authentication Failure  
• 5 = EGP Neighbor Loss  
• 6 = Enterprise Specific  
To send all traps received from the filtering mechanism to the defined nodes, set the bit mask to 254 or 0xFE. To send only Link Down (2) and Link Up (3) traps to the defined nodes, set the corresponding bits to “on” (0011 0000). To set the bit mask, enter the decimal value representation **48** or the hexadecimal value **0x30**. |
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Values</th>
<th>Purpose and Description</th>
</tr>
</thead>
</table>
| Activation Schedule | Any valid character string for activation time. | **Purpose:** Specifies the time and days of the week when the trap destination policy is to be activated and deactivated.  
Activation: Use this field to delay the onset of sending traps. Specify time in the format HH:MM, where HH is an integer in the range of 0–23 and MM is an integer in the range of 0–59. If this field is not set, the time is set to 00:00.  
Select the days of the week when you want the trap destination policy to be active.  
Deactivation Time: Use this field to set the time when the trap destination policy will be deactivated. Specify time in the format HH:MM, where HH is an integer in the range of 0–23 and MM is an integer in the range of 0–59. If this field is not set, the time is set to 00:00.  
Select the days of the week when you want the trap destination policy to be deactivated. |

**Analysis Policy Dialog Box**

To define an analysis policy, select **Analysis** using the Policy Type button on the Agent Policy Manager Configuration main menu. The following dialog box is displayed as shown in Figure 48 on page 258.
The fields in this dialog box have the following meanings:

*Table 26. Analysis Policy Dialog Box Fields*

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Values</th>
<th>Purpose/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Any unique ASCII character string.</td>
<td><strong>Purpose</strong>: Specifies the name of a particular analysis entry. <strong>Description</strong>: The value in the Name field is used as a label and an instance ID, identifying each field for this entry, and is appended to the corresponding object ID. The instance IDs for each field in the same entry are the ASCII values for each letter of the name. For example, a value of APP_TIME causes the instance ID for this entry to be 65.80.80.95.84.73.77.69, where each letter is an ASCII integer representation of APP_TIME.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Values</td>
<td>Purpose/Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>State</td>
<td>Valid values are: Disabled, Enabled, Invalid</td>
<td><strong>Purpose:</strong> Identifies the current activity state of the analysis entry.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Description:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Disabled</strong> means that the entry is valid, but it is not currently available for use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Enabled</strong> means that the entry is available for use, but the results are not stored.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When you click <strong>Delete</strong>, the State field is changed to <strong>invalid</strong>, which causes the selected entry to be deleted.</td>
</tr>
<tr>
<td>Description</td>
<td>Any valid character string.</td>
<td><strong>Purpose:</strong> Specifies the general purpose of the analysis operation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Description:</strong> The description character string can be as detailed as necessary. This field is for informational purposes. This field is not mandatory, but is recommended.</td>
</tr>
<tr>
<td>Poll Time</td>
<td>Valid values include an integer followed by one of these letters:</td>
<td><strong>Purpose:</strong> Indicates the time period that should elapse before the next polling operation is invoked.</td>
</tr>
<tr>
<td></td>
<td>d = days</td>
<td><strong>Description:</strong> The polling is done at the defined intervals and the retrieved values are cached for later use in evaluating expressions. Thus, no SNMP get request is necessary while an expression is being processed.</td>
</tr>
<tr>
<td></td>
<td>h = hours</td>
<td></td>
</tr>
<tr>
<td></td>
<td>m = minutes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>s = seconds</td>
<td></td>
</tr>
<tr>
<td></td>
<td>u = milliseconds</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The minimum is 1 minute.</td>
<td></td>
</tr>
<tr>
<td>MIB Variable Expression</td>
<td>Can be comprised of valid MIB object IDs, operators, and functions.</td>
<td><strong>Purpose:</strong> Indicates the mathematical expression to be evaluated on the retrieved MIB variable values.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Description:</strong> In MLM expressions, you can specify MIB variables from local and remote nodes. To use remote nodes, use the remote node alias name, host name, or IP address and a colon (:) as a prefix to the MIB object ID. You can also specify MIB variables from multiple instances by appending a wildcard character (*) to the end of the MIB object ID. In your expression, use operators from a predefined set or the provided built-in functions. See <strong>Operators for MIB Variable Expressions</strong>, <strong>Functions for MIB Variable Expressions</strong>, for descriptive lists of the available operators and functions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Click <strong>Select</strong> to access the Tivoli NetView MIB browser, which can help you obtain object IDs and other information about the MIB variables on which you want to perform analysis.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When MIB variables return multiple instance values, all resolved values are averaged before they are used in the expression.</td>
</tr>
</tbody>
</table>
Table 26. Analysis Policy Dialog Box Fields (continued)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Values</th>
<th>Purpose/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result Type</td>
<td><strong>Valid values are:</strong></td>
<td><strong>Purpose:</strong> Indicates the type of output request to be used for operation results.</td>
</tr>
<tr>
<td></td>
<td>Integer</td>
<td><strong>Description:</strong> Based on the Result Type value, the results for the command are saved to a predefined corresponding MIB variable. The results can be propagated to the other predefined MIB variables. For example, if the Result Type value is <code>integer</code>, when the expression is evaluated, the output is converted to an integer, saved in the IntegerResult MIB variable, and propagated to all other variables based on IntegerResult. Results for <code>integer</code> are signed mathematical values. Results for <code>counter</code> and <code>gauge</code> are unsigned mathematical values. The results for <code>counter</code> wrap if the value is over or under; the results for <code>gauge</code> latch at 0 (zero), if under and 4294967295 if over.</td>
</tr>
<tr>
<td></td>
<td>Counter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Gauge</td>
<td></td>
</tr>
</tbody>
</table>

**Alias Policy Dialog Box**

To define an alias policy, select **Alias** using the Policy Type button on the Agent Policy Manager Configuration main menu. The following dialog box is displayed as shown in [Figure 49 on page 261](#).
The fields in this dialog box have the following meanings:

Table 27. Alias Policy Dialog Box Fields

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Values</th>
<th>Purpose/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Any character string</td>
<td><strong>Purpose:</strong> Defines the alias name for the objects in the List field.</td>
</tr>
<tr>
<td></td>
<td>expression.</td>
<td><strong>Description:</strong> This alias can be used in other policies, such as</td>
</tr>
<tr>
<td></td>
<td></td>
<td>threshold, analysis, or trap destination to perform functions on</td>
</tr>
<tr>
<td></td>
<td></td>
<td>multiple remote nodes.</td>
</tr>
</tbody>
</table>

Figure 49. Alias Policy Dialog Box
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Values</th>
<th>Purpose/Description</th>
</tr>
</thead>
</table>
| State      | Valid values are: Enabled, Disabled, Invalid | **Purpose:** Indicates whether the alias is currently available for use.  
**Description:**  
- **Enabled** means that the alias entry is valid and available for use.  
- **Disabled** means that the alias entry is valid, but is dormant and is not currently being used.  
When you select **Delete**, the State field is changed to **invalid**, which causes the selected entry to be deleted. |
| List       | Valid values are: Alias names, Host names, IP addresses, Enterprise IDs | **Purpose:** Indicates the nodes represented by the alias.  
**Description:** Separate multiple values by a comma or space. You can group alias names under other alias names. Also, nodes can be grouped under multiple alias names to meet your needs. You might want to give an alias name to enterprise IDs for use in the Filter Table, enabling filtering by enterprise ID. |

**Administration Policy Dialog Box**

To define an administration policy, select **Administration** using the Policy Type button on the Agent Policy Manager **Configuration** main menu. The following dialog box is displayed as shown in Figure 50 on page 263.
The fields in this dialog box have the following meanings:

Table 28. Administration Policy Dialog Box Fields

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Values</th>
<th>Purpose and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Any unique ASCII character string.</td>
<td><strong>Purpose</strong>: Specifies the name of a particular entry in the table. <strong>Description</strong>: The value in the Name field is used as a label and an instance ID, identifying each field for this entry, and is appended to the corresponding object ID. The instance IDs for each field in the same entry are the ASCII values for each letter of the name. For example, a value of APP_TIME causes the instance ID for this entry to be 65.80.80.95.84.73.77.69, where each letter is an ASCII integer representation of APP_TIME.</td>
</tr>
</tbody>
</table>
Table 28. Administration Policy Dialog Box Fields (continued)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Values</th>
<th>Purpose and Description</th>
</tr>
</thead>
</table>
| State      | Valid values are: Valid, Outdated, Invalid | Purpose: Identifies whether the entry is current.  
Description:  
- **Valid** means that the information in this entry is authentic and can be used.  
- **Outdated** means that the information in this entry is obsolete and should not be used or should be used with this knowledge.  
When you select Delete, the State field is changed to invalid, which causes the selected entry to be deleted. |
| Description| Any valid character string. | Purpose: Indicates the general purpose of the information in the entry.  
Description: The description character string can be as detailed as necessary. This field is for informational purposes. This field is not mandatory, but is recommended. |
| Value      | Any valid ASCII string. | Purpose: Provides administrative information. |
| Owner      | Any valid ASCII string. | Purpose: Indicates the owner of this piece of information. |

Diagnosing Problems Using the Problem Determination Assistance Facility

APM provides a tool that can aid you in determining what the conditions were that triggered a threshold or file monitor trap or just display data about an APM policy. To see the Problem Determination Assistance dialog box, do the following:

1. Double-click **APM Monitors**.
   
   A map with the SmartSets used by APM is displayed.
2. Double-click a SmartSet icon to see the objects in the SmartSet.
3. Double-click down to the node-level view, which looks similar to Figure 51 on page 265.
If a threshold or file monitor condition was met, one of the icons is red. If an icon is blue, then either the distribution of the policy to the node failed or the session between the node and its managing MLM went down.

4. Double-click the icon.

Note that you can display the Problem Determination Assistance dialog box even if all the icons are green.

A Problem Determination Assistance dialog box, similar to Figure 52 on page 266, is displayed.
From this menu, you can see a summary of the policy settings, start some applications, perhaps to help you diagnose the problem that triggered the threshold or file monitor condition, and reset the color of file monitor icons. You can also create your own scripts to maintain log files, erase files, and do automated recovery. Commands run on the Tivoli NetView host.

### Resetting the Color of a File Monitor Icon

You can reset the color of a file monitor icon from the Problem Determination Assistance dialog box by clicking **Reset** in the middle of the dialog box. The color of the icon from which you started the Problem Determination Assistance dialog box is changed to green, and the change in status is propagated up to the root.
Starting Applications from the Problem Determination Assistance Dialog Box

The following diagnosis applications are provided on the Problem Determination Assistance dialog box:

- **Display submap** starts a submap showing the node-level view for the affected node. This view has two icons.
  - The node interface card, as on other Tivoli NetView maps.
  - One or more executable icons indicate thresholds and file monitor conditions that have been created.

- **Dynamic events for this node** opens a dynamic workspace showing all events for the node.

- **MIB Browser** starts the xnmbrowser application for this node.

- **Historic Graph Data** starts the xnmgrapher application and graphs the trap data that was stored for file monitors and thresholds against the node. The data is stored in the /usr/OV/databases/C5 directory.

By clicking on the **Add** button, you can add the following applications to the Problem Determination Assistance dialog box:

- **Historic events for this node** opens a static workspace showing all events for the node.

- **Print** starts the Tivoli NetView Print Tool application.

- **Mail** starts the mail application.

Select the application you want to start on the Problem Determination Assistance dialog box and then click **Start**.

Adding an Application to the Problem Determination Assistance Dialog Box

To add your own application to the Problem Determination Assistance dialog box, follow this procedure:

1. Click **Add** on the Problem Determination Assistance dialog box (when no applications are selected).

   The Actions List panel is displayed as shown in [Figure 53 on page 268](#).
2. Click Define/Edit.
   
   The Define/Edit dialog box is displayed as shown in Figure 54.

3. Enter the name for your application in the Description field and the full path name of the command or shell script that is to be executed in the Command Field and click OK. For example, to add an application named Display
Warning Pop, which executes the ovxecho command, enter **Display Warning Pop Up** in the Description field and the following commands in the Command field:

```
export DISPLAY=host1:0;/usr/0V/bin/ovxecho 'I have a problem'
```

Use a semicolon (;) to separate multiple commands as shown in Figure 54 on page 268.

The name of your application, Display Warning Pop Up, is added to the Application Actions section on the Problem Determination Assistance dialog box.

4. Select Display Warning Pop Up on the Problem Determination Assistance dialog box and select **Start** to execute your application.

### Example of Defining and Distributing a Policy

To help you understand how APM interacts with the SmartSet Facility, Tivoli NetView product, and Tivoli NetView MLM, read this step-by-step scenario of how an administrator would define a SmartSet and set up a file monitoring condition.

1. A network administrator at a university has been having problems with someone logging into a file server in the network and tampering with files to which only root has access. On the assumption that the troublemaker is guessing at root passwords, and thus making several faulty login attempts each time a machine is broken into, the administrator decides to monitor the `/etc/security/failedlogin` file for failed attempts. He will use the file monitoring capability provided by the Tivoli NetView MLM SIA, which is installed on each of the file servers. Since the `/etc/security/failedlogin` file is in binary, the administrator will use the `fwtemp` accounting command provided on the operating system to translate the file to ASCII before file monitoring begins.

2. First, the administrator defines a SmartSet called **fileservers** that includes the important nodes, starts the SmartSet Editor from the Tivoli NetView Tools menu and clicks **Add** to add a new SmartSet. On the Add SmartSet dialog, the administrator defines a SmartSet called **fileservers**, using the Node List definition type to specify the IP addresses of the operating system file servers. Click **OK**.

3. Next, the administrator selects **APM Configuration** from the **Tools** pull-down menu and clicks **Add/Copy** to add a file monitoring condition.

4. On the File Monitor dialog box as shown in Figure 55 on page 270, the administrator defines the following entries:
The file monitor entry Badlogins will check the /tmp/failedlog.ascii file for the string root every five minutes.

The administrator clicks on **Automated Actions**. He defines the following entries in the Automated Actions dialog box as shown in Figure 56 on page 271.
The command `/usr/sbin/acct/fwtmp < /etc/security/failedlogin` 
/tmp/failedlog.ascii uses the fwtmp utility to convert the binary file to ASCII and put the translated information into a temporary file called failedlog.ascii.

After this command is executed, file monitoring begins. Every five minutes, the file is tested for the string `root`. If the string is found, a special file monitor trap is sent to the Tivoli NetView product.

By default, the SIA would mark the file at the location where it stopped monitoring and would resume monitoring at that point in the file. In this particular installation, that default has been changed to reset monitoring to the beginning of the file between polling intervals. To prevent the SIA from generating another trap in the next polling interval for the string that was already found in the file, the administrator specifies the command `grep -v root /tmp/failedlog.ascii | /usr/sbin/acct/fwtmp -ic > /etc/security/failedlogin` as the command to be executed after the string is found. This command takes all contents of the failedlog.ascii file except the failed root login and writes the contents back into the failedlogin file. During the next polling cycle, when the binary file is again converted into ASCII, it will not have the records that caused the previous file monitor trap to be sent.

5. To specify SmartSets to which this file monitoring policy will be applied, the administrator clicks Assign on the File Monitoring dialog box. On the SmartSet Assignments dialog that is displayed, the administrator selects Fileservers and then clicks Assign. Fileservers is moved to the Assigned SmartSets list, and the administrator clicks OK to make the assignment. (Starting the SmartSet Editor and defining the SmartSet from this dialog if it was not already defined, is an option.)

6. On the File Monitor dialog box, the administrator clicks Apply to apply the setting. A message is displayed in the Messages area that indicates the policy was saved successfully.

7. Now the administrator needs to distribute the policy to the machines that make up the fileservers collection. From the APM main dialog, he selects the badlogins policy and clicks Distribute.

8. On the Distribute Definitions dialog that is displayed, click Start to distribute the collection. Messages indicate when the set to each node is complete. (If one or more of the sets had failed, a message indicating why it failed is displayed.) The policy status will be partially distributed.

9. The administrator goes back to the Tivoli NetView Control Desk and selects Create -> Dynamic Workspace to start a dynamic workspace. On the dialog that is displayed, he selects Filter Activation in the Filter section of the dialog. On the dialog box that is displayed, select File List to get a list of filters. From
the list, he selects **C5filters** and clicks **OK**. In the Available Filters in File section of the dialog box, he selects **APMConsole** and **Activate**. He then clicks **Close** and **OK** to open the workspace.

10. Later that afternoon, the SmartSets icon on the root map turns yellow. The administrator quickly checks the dynamic workspace set up earlier and sees that file monitor specific trap 21 (String Found) arrived. The perpetrator has been caught in the process of breaking into one of the file servers.

11. Some time later (after locating the guilty party in the university computing center), the administrator double-clicks down through the APM Monitors submaps to the node-level view map for the affected node. Double-clicks the executable icon to display the Problem Determination Assistance dialog box. Resets the status on the node so that the color returns to green.

---

### APM Reference

The following sections have more detailed information about the APM and how it interacts with Tivoli NetView MLM and the network.

#### Configuring Community Names

The APM application uses SNMP community names to control the access the Tivoli NetView manager has to managed objects in the network. Community names are passed back and forth in SNMP get and set requests and in traps. The community name is similar to a password in that it determines whether an entity can gain access to information or perform an action.

On SNMP-based network devices, SNMP community names are defined in a file called snmpd.conf. The SNMP daemon checks this file when it receives a request for information.

The Tivoli NetView product and its associated applications, such as Tivoli NetView MLM, use another file called ovsnmp.conf to associate objects with community names. The Tivoli NetView product and Tivoli NetView MLM check this file when they send requests for information.

In summary:

- When a node *sends* a request, it uses the ovsnmp.conf file to determine what community name should be sent in the request. The community name associated with the target node is included in the request.
- When a node *receives* a request, it uses the snmpd.conf file to validate the community name sent by the requester. The community name associated with the requesting node is compared with the name included in the request.

The community name sent by the requester node and the community name expected by the receiving node *must match* for the request to be executed as requested. If they do not match, the receiving node will use a community name of **public** by default.

Mismatched community names cause authentication errors. These errors prevent the distribution of an APM file monitor or threshold policy from being successful.

You should not edit the ovsnmp.conf file directly. Select **SNMP Configuration** from the **Options** pull-down menu to set the community name to be sent to the target node.
If you are defining a threshold condition in your network, your Tivoli NetView manager station must have SNMP set access to the nodes running MLMs. For this reason, it might be easier to use a single community name in your network. The Tivoli NetView security feature can provide the application security that you require.


Community Name Examples
Here are two scenarios of how entries in the ovsnmp.conf and snmpd.conf files change, depending on which objects you are monitoring. For simplicity, these examples show a single machine running each agent. In a real working network, community names are be set up on a much wider basis, such as across a subnet or even across the whole network.

First Example
The Tivoli NetView product, an MLM, and an SIA are on a machine called nvmgr. You want to set a thresholding condition against that MLM and monitor a log file using the SIA file monitoring function.

- The ovsnmp.conf file is edited using the SNMP Configuration dialog from the Tivoli NetView Options menu:

```
# MLM entry (LOOPBACK)
127.0.0.1:system:*:::::system:
# SIA entry
nvmgr:siaset:*:::::siaset:
```

- The snmpd.conf file entries are entered as follows:

```
# MLM name
community system 127.0.0.1 255.255.255.255 readWrite
# SIA name
community siaset 127.0.0.1 255.255.255.255 readWrite
```

Second Example
An MLM and an SIA are on a machine called workstation. From the nvmgr node, you want to set a thresholding condition against that MLM and monitor a log file using the SIA file monitoring function.

- The ovsnmp.conf file is edited using the SNMP Configuration dialog from the Tivoli NetView Options menu:

```
# MLM entry
workstation:mlmset:*:::::mlmset:
# SIA entry
workstation:siaset:*:::::siaset:
```

- The snmpd.conf file entries are entered as follows:

```
# MLM name
community system 127.0.0.1 255.255.255.255 readWrite
# SIA name
community siaset 127.0.0.1 255.255.255.255 readWrite
```

See the Tivoli NetView for UNIX Mid-Level Manager User’s Guide for extensive scenarios showing how you would configure community names depending on where you have the Tivoli NetView MLM installed.
APM Aliases

You can use APM to customize an Alias Table that is used for several MLM functions.

Managing Aliases

If the administrator configures the MLMs in the network to take over local discovery and status monitoring duties from the Tivoli NetView manager, the MLM assigns aliases for groups of nodes to facilitate management of the nodes. APM also uses the Alias Table to keep track of groups of nodes. It sets aliases for SmartSets that have thresholds set against them, and for the thresholds themselves.

After you define a threshold and distribute it to a SmartSet, an alias is defined for this SmartSet. Aliases for SmartSets have names in the format \{NV_ip-address_ssname\}, where ip-address is the IP address of the Tivoli NetView manager, and ssname is the name of the SmartSet. For example, a SmartSet called Fileservers assigned to the MLM by the Tivoli NetView manager on address 9.67.102.16 would be assigned an alias of \{NV9.67.102.16_Fileservers\}.

Similarly, there is an alias for each threshold you define. Thresholds have aliases in the format \{NV_ip-address_thresholdname\}, where ip-address is the IP address of the Tivoli NetView manager, and thresholdname is the name of the threshold. Do not update these aliases with the Tivoli NetView MLM Configuration Interface.

The alias policies are APM’s mechanism for managing changes to SmartSets and thresholds. SmartSets are dynamically updated to reflect changes in a network. When a SmartSet is changed, the change is made to the alias policy as well. You do not have to redistribute a threshold policy if you use the SmartSet Editor to change the members of a SmartSet; the APM automatically redistributes, based on the changes to the alias.

APM MLM Domains

For thresholding, APM defines domains for the MLMs in the network. Basically, it determines where the MLMs are and divides the managed objects as equitably as possible.

Initially, MLM domains are defined based on subnet IDs. Nodes that are in the same subnetwork as an MLM are assigned to that MLM. Nodes that do not have an MLM in their subnet are assigned to a default domain. There is no overlap in these initial policies; each node is managed by one MLM only. If there are several MLMs in a subnet, then the first one found manages the nodes in the subnet, and the other MLMs only manage themselves.

To see these domains from the Tivoli NetView root map, double-click the MLM Managers icon. Also, if you start the SmartSet Editor, you will see the MLM domains (as well as the mlmDomain_Default domain) listed as already existing SmartSets.

APM chooses the MLM that will serve as the default domain depending on how your network is set up:
1. If there is an MLM on the Tivoli NetView manager, it will be automatically assigned responsibility for the default domain.
2. If the MLM is not installed on the Tivoli NetView manager, you must assign the default domain SmartSet to one of the MLMs in your network using the SmartSet Editor.
If there is only one MLM in your network, that MLM is assigned all nodes, and a default domain might not be created.

**Rearranging MLM Domains Manually**

You might want to alter the distribution of nodes to MLMs to facilitate your management of the network. For example, you might want an MLM to manage all the routers in the network, or to manage all objects in a physical location. To do this, start the SmartSet Editor and select the domain you want to change, then edit the list of nodes. Because the MLM domain is just a SmartSet, you can use the same rule logic that you use for defining SmartSets to set thresholds against. You do not need to reassign any nodes that were in the old domain but are not in the new domain; APM takes care of assigning new domain responsibilities for the nodes. APM puts the nodes in the default domain or the domain for the MLM for the subnet.

**Note:** When you remove subnet responsibility from an MLM, you should assign the responsibility to another MLM in the subnet. If you do not, the entire subnet will be placed in the default domain.

The netmon daemon and APM can share the mlmDomain collections for work distribution if you configure the netmon daemon to do so. See the netmon man page for more information.

If you have MLMs that are offloading discovery and status monitoring from the Tivoli NetView product and are very busy as a result, you might want to lighten their workload by moving nodes out of their APM domains.

If you add a new MLM to your network, APM recognizes it as an MLM and assigns a domain SmartSet, but the SmartSet contains only the MLM. If you edit this SmartSet and add nodes to it, APM takes care of moving node responsibilities accordingly.

**Automatic MLM Backup Domains**

If an MLM disappears from the network, APM automatically redistributes that MLM’s workload. A new SmartSet is created called delDomain_<mlmname>, where <mlmname> is the name of the MLM. The nodes for which the MLM was responsible are redistributed to other MLMs in the network. They are assigned according to their subnets.

If the MLM later reappears in the network, the domain is recreated and the nodes are reassigned to the original MLM.

**Matched, Armed, and Disarmed Command Environment Variables**

The commands entered in the Matched Command, Armed Command, and Disarmed Command fields on the Threshold Policy and Filter Policy dialog boxes can use the environment variables that contain information from the received trap.

Table 29 contains information about the MLM and the environment in which it runs:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM6K_HOSTNAME</td>
<td>Contains the local host name of the node on which the MLM is running.</td>
</tr>
</tbody>
</table>
Table 29. MLM Environment Variables (continued)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM6K_DOMAIN_NAME</td>
<td>Contains the local domain name, or fully qualified host name, on which the MLM is running.</td>
</tr>
<tr>
<td>SM6K_HOST_ADDRESS</td>
<td>Contains the IP address of the host on which the MLM is running.</td>
</tr>
<tr>
<td>SM6K_INSTANCE_NAME</td>
<td>Contains the name of the filter rule that matched the trap.</td>
</tr>
<tr>
<td>SM6K_INSTANCE_ID</td>
<td>Contains the instance ID of the filter rule that matched the trap.</td>
</tr>
<tr>
<td>SM6K_EXECUTION_REASON</td>
<td>Indicates why the command is being executed. The possible values are Filter_Matched, Filter_Armed, or Filter_Disarmed:</td>
</tr>
<tr>
<td></td>
<td>• Filter_Matched means the command is being entered in the Matched Command field on the Filter Table window. The command will be executed by the MLM when a received trap matches the filter rule.</td>
</tr>
<tr>
<td></td>
<td>• Filter_Armed means the command is being entered in the Armed Command field on the Throttle window. The command will be executed by the MLM when the number of received matching traps matches the value for the Arm Count, and the throttle is armed.</td>
</tr>
<tr>
<td></td>
<td>• Filter_Disarmed means the command is being entered in the Disarmed Command field on the Throttle window. The command will be executed by the MLM when the currently armed throttle is disarmed. The throttle can be disarmed, because the number of received matching traps matches the value for the Disarm Count or the time period specified in the Disarm Timer has elapsed.</td>
</tr>
</tbody>
</table>

Table 30 shows the environment variables that are set to reflect values from the received trap:

Table 30. Environment Variables Reflecting Information Received in Traps

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM6K_TRAP_ENTERPRISE</td>
<td>Contains the enterprise ID contained in the trap.</td>
</tr>
<tr>
<td>SM6K_TRAP_AGENT_ADDRESS</td>
<td>Contains the IP address of the agent sending the trap.</td>
</tr>
<tr>
<td>SM6K_TRAP_GENERIC_NUM</td>
<td>Contains the generic trap number of the trap.</td>
</tr>
<tr>
<td>SM6K_TRAP_SPECIFIC_NUM</td>
<td>Contains the specific trap number of the trap.</td>
</tr>
<tr>
<td>SM6K_TRAP_TIME_TICKS</td>
<td>Contains the time ticks since the agent generating the trap was started.</td>
</tr>
<tr>
<td>SM6K_NUM_TRAP_VARS</td>
<td>Contains the number of MIB variables contained in the trap.</td>
</tr>
</tbody>
</table>
The following environment variables correspond to the MIB variables contained in the trap. These environment variables are set for each variable in the trap. To differentiate between these environment variables for the different MIB variables, the positional number is appended to the environment variable. MIB variable numbering ranges from 1 to $n$, where $n$ corresponds to the number of MIB variables contained in the trap.

The environment variables that correspond to the MIB variables in the received trap are listed in Table 31:

**Table 31. Environment Variables Reflecting MIB Variables from Received Traps**

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM6K_TRAP_VAR_OBJ_ID#</td>
<td>Contains the object ID for MIB variable number # in the variable bindings.</td>
</tr>
</tbody>
</table>
| SM6K_TRAP_VAR_TYPE# | Indicates the type of the MIB variable number #. The type is one of the following:  
  - INTEGER  
  - Counter  
  - Gauge  
  - TimeTicks  
  - IpAddress  
  - OCTET_STRING  
  - Opaque  
  - OBJECT_IDENTIFIER  
  - NULL  
  - Unknown |
| SM6K_TRAP_VAR_VALUE# | Contains the value of the MIB object. For the MIB types INTEGER, Counter, Gauge, and TimeTicks the value is a decimal integer. For the MIB type IpAddress, the value is stored in dot notation. For the MIB types OCTET_STRING and Opaque, the value is stored as an ASCII string unless it contains unprintable characters. If it contains unprintable characters, it is stored as a hexadecimal string. For MIB type OBJECT_IDENTIFIER, the value is stored using dot notation. For MIB type NULL, the value is set to NULL. |

The environment variables described in the following table are valid for Armed and Disarmed commands. These environment variables correspond to specific filter throttle MIB variables, as shown in Table 32:

**Table 32. Environment Variables Reflecting Filter Throttle MIB Variables**

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM6K_THROTTLE_ARM_COUNT</td>
<td>Contains the value of the ThrottleArmTrapCount MIB variable (the Arm Count value).</td>
</tr>
<tr>
<td>SM6K_THROTTLE_DISARM_COUNT</td>
<td>Contains the value of the ThrottleDisarmTrapCount MIB variable (the Disarm Count value).</td>
</tr>
<tr>
<td>SM6K_EXECUTION_REASON</td>
<td>Set to Filter_Armed when the throttle is armed, or set to Filter_Disarmed when the throttle is disarmed.</td>
</tr>
<tr>
<td>SM6K_THROTTLE_DISARM_TIMER</td>
<td>Contains the string value of the ThrottleDisarmTimer MIB variable (the Disarm Timer value).</td>
</tr>
</tbody>
</table>
Operators for MIB Variable Expressions

The MIB variable expression that you specify in an analysis policy can be assigned the following operators. The expressions are evaluated in order of precedence and associativity of operators:

- Grouping
- Unary minus
- Multiplication
- Division
- Remainder
- Addition
- Subtraction
- Bitwise left shift
- Bitwise right shift
- Bitwise AND
- Bitwise exclusive OR
- Bitwise inclusive OR

Functions for MIB Variable Expressions

The MIB variable expression that you specify in an analysis policy can use the following functions:

Notes:

1. An MLM MIB variable can have an IP address, host name, or alias appended to it and can resolve to multiple values.
2. All wildcard MIB variables are averaged before being used.

@sum(MIB variable,MIB variable,)
   Returns the sum of all values.

@avg(MIB variable,MIB variable,)
   Returns the average of all values.

@min(MIB variable,MIB variable,)
   Returns the minimum of the values.

@max(MIB variable,MIB variable,)
   Returns the maximum of the values.

@diff(MIB variable expression,MIB variable expression)
   Returns the positive difference of the two expressions. This function is useful for counters that are always positive.

@abs(MIB variable expression)
   Returns the absolute value of the expression.

@delta(MIB variable expression)
   Uses the delta value calculated by subtracting the last returned MIB variable value from the current MIB variable value. This function can be used to force delta calculations on integer, gauge, and displaystring MIB variables.

@value(MIB variable expression)
   Uses the current MIB variable value. This function can be used to force the use of actual values for counter MIB variables.

@rand()
   Returns a random number between (0) and MAX_INT.

@numInst(regular expression)
   Returns the number of instances found for the expression. This function
enables you to monitor a table for a change in the number of instances and
to monitor the process table for same-name processes to see whether any is
removed or added.
Appendix A. Tivoli NetView Internal Traps

Use this reference to understand the traps that are generated by the Tivoli NetView product.

Terms and Conventions

The following terms and conventions are used:

- Words in italics are explanatory variable names. They are replaced by the value of the variables at run time when the event occurs.
- The term *event* is used interchangeably with the term *trap*.
- The term event log generally implies either the trapd.log file or the event card application display.
- Time stamp values are displayed in epoch time. Epoch time is the number of seconds elapsed since an epoch, which in UNIX is January 1, 1970.
- Some trap names are preceded by either IBM_ or IBM_NV in the Tivoli NetView for UNIX trapd.conf file. Tivoli NetView for Windows uses different trap names and these are noted in Table 33 on page 284.
- This appendix uses the UNIX convention for specifying directory paths. For Windows platforms, replace each forward slash (/) with a backslash (\) and the uppercase OV with lowercase ov in directory paths. For example, change the /usr/OV/conf/ovsnmp.conf path to \usr\ov\conf\ovsnmp.conf.
- The term objid is an OVw database object ID.
- Link Level Address, (LLA) is the same as physical address which is the physical address of the interface cards. The LLA is typically displayed as a 6 byte, (or 12-digit hexadecimal) number.
- When a demand poll is used as a method of correction, it should be pointed out that a poll is, by default, performed once-a-day for each node. Performing a demand poll is a method of effecting immediate action.

**Note:** A demand poll can be performed from a separate window by the command `nmdemandpoll nodename` where node name is the node name as it exists in the topology database (generally, the fully qualified DNS name for the node).

Internal Tivoli NetView Traps

The following traps are the only traps listed in Table 33 on page 284 that are not generated by the Tivoli NetView product itself:

- The SNMP_EV (58916871) trap, which can be generated by the user to facilitate Configurable Status.
- The DISPSUB (59179073) trap, which can be sent by the Tivoli Event Console back to the Tivoli NetView product to display a submap or to launch a Tivoli NetView application.

All Tivoli NetView traps are generated with the enterprise ID \1.3.6.1.4.1.2.6.3.1 and the generic number of 6, which means enterprise-specific. The specific number for each trap is listed in Table 33 on page 284.
A Tivoli NetView trap is generated with up to 10 variable bindings. The OID for each varbind is also listed, though it has relatively no importance in the generated trap.

**Varbind 1**

MIB OID: .1.3.6.1.4.1.2.6.3.1.1.2.0

This variable is the source ID. It is an integer value that corresponds to the internal component of the Tivoli NetView product that generated the event. Following are the separate Source ID values and their corresponding components. Each source ID value has a unique letter (identified in parenthesis) that is used for identification in the trapd.log file. For instance, the Node Up event text is preceded by an N, indicating that the source for the trap was the netmon application.

- (A)gent
- (n)etmon related
- (a)pplication
- (O)SI_SuperAgent
- (D)ata Collector
- (M)ap/ovtopmd
- (d)emo/LoadHosts
- (S)ecurity Agent
- (E)vent Application
- (s)spappld
- (I)PMAP_SuperAgent
- (T)rapd
- (i)gnore
- (r)alertd
- (L)oad MIB
- (V)endor
- (N)etmon
- xnm(C)ollect
- (NonI(P) Topology
- xnm(t)rap

**Varbind 2**

MIB OID: .1.3.6.1.4.1.2.6.3.1.1.3.0

This variable is a string value identifying the host name to which this trap applies. If there is no applicable node, <none> is displayed. As in our previous varbind example, if the Node Up event is generated, then the varbind-2 variable will contain the host name of the node that was detected as operational.

**Varbind 3**

MIB OID: .1.3.6.1.4.1.2.6.3.1.1.4.0

This variable is a string value containing a description of the event that was generated. For the Node Up event, varbind-3 would contain the text Node Up. This varbind is most important because the third varbind variable is the text that gets displayed in the event log. This is most evident by looking at the Event Configuration screen of the product. For each Tivoli NetView event, the corresponding event log information contains $3. This indicates that the contents of the third varbind are to be displayed in the event log. The exact text for each trap is documented in Table 33 on page 284.

**Varbind 4**

MIB OID: .1.3.6.1.4.1.2.6.3.1.1.5.0

This variable is a string value containing internal data that is particular to the trap. This field generally contains data such as timestamp values, object ID values for node and interface objects, IP addresses, and so forth. This information generally has no meaning to the user. The exact value for each trap is documented in the tables.

**Varbind 5**

MIB OID: .1.3.6.1.4.1.2.6.3.1.1.6.0

This variable contains a string value containing the database name. This name must be openview.
Varbind 6
MIB OID: .1.3.6.1.4.1.2.6.3.1.1.7.0
This variable is used infrequently, but when it is used it is the object
selection name.

Varbind 7
MIB OID: .1.3.6.1.4.1.2.6.3.1.1.8.0
For Subnet Unreachable and Subnet Reachable Again events, this variable
contains the network address of the subnet.
For all interface events this variable contains the IP address of the
interface.
For all node events, this variable is not used.
For all router events, this variable is not used.
For all Layer 2 Status events, this variable is not used.

Varbind 8
MIB OID: .1.3.6.1.4.1.2.6.3.1.1.9.0
For Subnet (network) Unreachable and Subnet (network) Reachable Again
events, this variable contains the subnet mask.
For all interface events, this variable contains the interface name. The
interface name is the short name assigned to an interface (for example,
eth0 or hme0). This is used to uniquely identify the interface on a given
node
For all node events, this variable contains a comma-separated list of the IP
addresses of all the interfaces on the node.
For all router events, this variable is not used.
For all Layer 2 Status events, this variable contains a comma-separated list
of IP addresses for all interfaces on the node.

Varbind 9
MIB OID: .1.3.6.1.4.1.2.6.3.1.1.10.0
This variable is only used for Layer 2 Status events. It contains the subnet
IP address.

Varbind 10
MIB OID: .1.3.6.1.4.1.2.6.3.1.1.11.0
This variable is only used for Layer 2 Status events. It contains the subnet
mask.

 Trap list
The traps are listed in numerical order. For each trap listed, the following
information is provided:
• Specific trap number
• Trap name
• Trap description
• Description field (descr): This contains the contents of varbind-3.
• Data field: This contains the contents of varbind-4.
• Condition: This describes the reason that the trap is generated.

Traps generated by the Tivoli NetView product are described as follows.
<table>
<thead>
<tr>
<th>Number, UNIX name, Windows name, and Description</th>
<th>Fields and Condition</th>
</tr>
</thead>
</table>
| 50462720  
WARN_EV  
IBMWARN  
Warnings | **Descr field**  
It can be one of the following strings. The number corresponds with the number under the Condition field.  
1. WARNING Attempted addition of object with no object ID, *objectName*  
2. WARNING Attempted addition of existing object *objectId*, returning error  
3. WARNING could not allocate object ID for segment *segName*  
4. WARNING Unset field values with object id *oid* failed: OVwError = *errorString*.  
5. WARNING invalid SNMPTrap packet from agent *addr* source *sourceId* pid *pid*  
**Data field**  
NULL  
**Condition**  
Following are the conditions:  
1. Attempt was made to add an object with null object ID to the topology database.  
2. Attempt was made to add an already existing object to the topology database.  
3. Creation of object ID for segment fails.  
4. Unsetting of field values in the topology database fails during cleanup.  
5. Error parsing an SNMP trap from an agent. |
| 50790400  
NM_EV  
NodeMarginal  
Node Marginal | **Descr field**  
Node Marginal  
**Data field**  
Contains the following:  
• Time stamp  
• objid of the node object  
**Condition**  
Node status change to a marginal state. |
| 50790401  
SN_EV  
SegNormal  
Segment Normal | **Descr field**  
Segment *segName* Up.  
**Data field**  
Contains the following:  
• Time stamp  
• objid of the segment object  
**Condition**  
Segment status changes to normal. |
Table 33. Tivoli NetView Internal Traps (continued)

<table>
<thead>
<tr>
<th>Number, UNIX name, Windows name, and Description</th>
<th>Fields and Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>50790402</strong>&lt;br&gt;SM_EV&lt;br&gt;SegMarginal&lt;br&gt;Segment Marginal</td>
<td><strong>Descr field</strong>&lt;br&gt;Segment segName Marginal</td>
</tr>
<tr>
<td><strong>Data field</strong>&lt;br&gt;Contains the following:&lt;br&gt;• Time stamp&lt;br&gt;• objid of the segment object</td>
<td><strong>Condition</strong> Segment status changes to marginal.</td>
</tr>
<tr>
<td><strong>50790403</strong>&lt;br&gt;NETN_EV&lt;br&gt;NetNormal&lt;br&gt;Network Normal</td>
<td><strong>Descr field</strong>&lt;br&gt;Network netName Up</td>
</tr>
<tr>
<td><strong>Data field</strong>&lt;br&gt;Contains the following:&lt;br&gt;• Time stamp&lt;br&gt;• objid of the network object</td>
<td><strong>Condition</strong> Network status changes to normal.</td>
</tr>
<tr>
<td><strong>50790404</strong>&lt;br&gt;NETM_EV&lt;br&gt;NetMarginal&lt;br&gt;Network Marginal</td>
<td><strong>Descr field</strong>&lt;br&gt;Network netName Marginal</td>
</tr>
<tr>
<td><strong>Data field</strong>&lt;br&gt;Contains the following:&lt;br&gt;• Time stamp&lt;br&gt;• objid of the network object</td>
<td><strong>Condition</strong> Network status changes to marginal.</td>
</tr>
<tr>
<td><strong>50790405</strong>&lt;br&gt;SA_EV&lt;br&gt;SegAdded&lt;br&gt;Segment Added</td>
<td><strong>Descr field</strong>&lt;br&gt;Segment segName Added</td>
</tr>
<tr>
<td><strong>Data field</strong>&lt;br&gt;Contains the following:&lt;br&gt;• Time stamp&lt;br&gt;• objid of the segment object</td>
<td><strong>Condition</strong> A segment was added.</td>
</tr>
<tr>
<td><strong>50790406</strong>&lt;br&gt;SD_EV&lt;br&gt;SegDeleted&lt;br&gt;Segment Deleted</td>
<td><strong>Descr field</strong>&lt;br&gt;Segment segName Deleted</td>
</tr>
<tr>
<td><strong>Data Field</strong>&lt;br&gt;Contains the following:&lt;br&gt;• Time stamp&lt;br&gt;• objid of the segment object</td>
<td><strong>Condition</strong> A segment was deleted.</td>
</tr>
<tr>
<td>Number, UNIX name, Windows name, and Description</td>
<td>Fields and Condition</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>---------------------</td>
</tr>
</tbody>
</table>
| **50790407**  
NETA_EV  
NetAdded  
Network Added | **Descr field**  
Network *netName* Added.  
**Data field**  
Contains the following:  
- Time stamp  
- objid of the network object  
**Condition**  
A network was added. |
| **50790408**  
NETD_EV  
NetDeleted  
Network Deleted | **Descr field**  
Network *netName* Deleted.  
**Data field**  
Contains the following:  
- Time stamp  
- objid of the network object  
**Condition**  
A network was deleted. |
| **50790411**  
CPP_EV  
ChgPollPeriod  
Change Polling Period | **Descr field**  
Polling intervals changed  
**Data field**  
NULL  
**Condition**  
Polling intervals were changed using the Topology/Status Polling Intervals window. This trap informs netmon that it needs to reread the polling intervals file. |
| **50790412**  
FP_EV  
ForcedPoll  
Forced Poll Event | **Descr field**  
Demand polling on node *nodename*  
**Data field**  
Contains the following:  
- *pipename*  
The pipe opened for communication with netmon  
- *debugflag*  
Reserved for future use  
**Condition**  
When a demand-poll is initiated by nmdemandpoll. The application communicates to netmon by sending this event. |
| **50790413**  
CFP_EV  
CnclForcdPoll  
Cancel Demand Poll | **Descr field**  
Cancel forced poll on node *nodename*  
**Data field**  
Contains the following:  
- *pipename*  
The pipe opened for communication with netmon  
**Condition**  
A demand poll operation (a forced poll) has been cancelled by the user. |
Table 33. Tivoli NetView Internal Traps (continued)

<table>
<thead>
<tr>
<th>Number</th>
<th>UNIX name</th>
<th>Windows name</th>
<th>Description</th>
<th>Fields and Condition</th>
</tr>
</thead>
</table>
| 50790416 | MNET_EV  | ManageNet | Manage Network | **Descr field**  
Network *netname* Managed  
**Data field**  
Contains the following:  
• Time stamp  
• objid of the network object  
**Condition**  
The event is generated when a network is managed. |
| 50790417 | UNET_EV  | UnmanageNet | Unmanage Network | **Descr field**  
Network *netname* Unmanaged  
**Data field**  
Contains the following:  
• Time stamp  
• objid of the network  
**Condition**  
The event that is generated when a network is unmanaged. |
| 50790418 | MN_EV    | ManageNode | Manage Node | **Descr field**  
Node Managed  
**Data field**  
Contains the following:  
• Time stamp  
• objid of the node  
**Condition**  
This event is generated when a node is managed. |
| 50790419 | UN_EV    | UnmanageNode | Unmanage Node | **Descr field**  
Node Unmanaged  
**Data field**  
Contains the following:  
• Time stamp  
• objid of the node  
**Condition**  
This event is generated when a node is unmanaged. |
| 50790420 | MSEG_EV  | ManageSeg | Manage Segment | **Descr field**  
Segment *segname* Managed  
**Data field**  
Contains the following:  
• Time stamp  
• objid of the segment  
**Condition**  
This event is generated when a segment is managed. |
<table>
<thead>
<tr>
<th>Number</th>
<th>UNIX name</th>
<th>Windows name</th>
<th>Description</th>
<th>Fields and Condition</th>
</tr>
</thead>
</table>
| 50790421 | USEG_EV  | Unmanage Seg | Unmanage Segment | Descri field: Segment *segname* Unmanaged  
Data field: Contains the following:  
- Time stamp  
- Objid of the segment  
Condition: This event is generated when a segment is unmanaged. |
| 50790423 | NMTM_EV  | ChgNMTrcMask | Netmon Change Trace Mask | Descri field: Netmon trace mask changed.  
Data field: *tracemask*  
Condition: If the `-M` option is used to change the netmon tracemask, this trap is generated to inform the netmon daemon of the new tracemask value. |
| 50790427 | CIS_EV    | ChgIntfSeg   | Change Interface Segment | Descri field: One of the following fields:  
- Interface *iflabel* no longer connected to segment.  
- Interface *iflabel* transferred to segment *segName*  
- Interface *iflabel* Transferred to segment *segId*  
Data field: Contains the following:  
- IP address  
- Time stamp  
- Node objid  
- Interface objid  
- Segment objid  
Condition: A change of interface event. |
| 50790429 | ChgSnmpOpts | Windows only | SNMP Change | Descri field: Changed *ovsnmp.conf* file  
Data field: Null  
Condition: This event is generated when the SNMP configurator updates the /usr/OV/conf/ovsnmp.conf file. |
| 50790437 | SNMPCHG_EV | UNIX Only | SNMP change | Descri field: *applicationName* changed format file /usr/OV/conf/ovsnmp.conf  
Data field: Null  
Condition: This event is generated when the SNMP configurator updates the /usr/OV/conf/ovsnmp.conf file. |
<table>
<thead>
<tr>
<th>Number, UNIX name, Windows name, and Description</th>
<th>Fields and Condition</th>
</tr>
</thead>
</table>
| 50790438 FMTCHG_EV ChgTrpFmtFile trapd.conf file format change | **Descr field**
programe changed format file /usr/OV/conf/C/trapd.conf.
**Data field**
NULL
**Condition**
This event is generated when contents of the trapd.conf are
changed either by the graphical interface application, xnmtrap, or
the command line interface, addtrap. This informs any interested
applications that they need to read the trapd.conf file again. |
| 50790439 MIBCHG_EV MIBFileChange MIB definition file format changed | **Descr field**
New MIB library file.
**Data field**
NULL
**Condition**
This event is generated when a MIB is loaded with the MIB
Loader. This informs any interested applications that a new MIB
binary file is available. |
| 50790440 COLCHG_EV ChgColCfgFile SNMP data collector file format changed | **Descr field**
SNMP data collector started.
**Data field**
xnmcollect changed format file /usr/OV/conf/snmpCol.conf
**Condition**
This event is generated when the SNMP Data collector
configuration file is updated. This informs the snmpCollect
domain that it should read its configuration file again. |
| 50790441 MI_EV ManageIntf Manage Interface | **Descr field**
Interface iflabel managed.
**Data field**
Contains the following:
• IP address
• Time stamp
• Node and interface objids
**Condition**
This event is generated when an interface is managed. |
| 50790442 UI_EV UnmangeIntf Unmanage Interface | **Descr field**
Interface iflabel unmanaged.
**Data field**
Contains the following:
• IP address
• Time stamp
• Node and interface objids
**Condition**
This event is generated upon when an interface is unmanaged. |
<table>
<thead>
<tr>
<th>Number, UNIX name, Windows name, and Description</th>
<th>Fields and Condition</th>
</tr>
</thead>
</table>
| 50790443 NETFC_EV NetFlagsChgd Network Flags changed | **Descr field**  
Network netName has new flags: netFlags.  
**Data field**  
Contains the following:  
- Time stamp  
- Network object ID  
- New network flags  
- Old network flags  
**Condition**  
This event is generated when the following internal flags for a network are changed:  
REMOVED USER_ADDED LAYOUT_OFF  
USE_XY SERIAL_NETWORK |
| 50790444 SEGFC_EV SegFlagsChgd Segment Flags changed | **Descr field**  
Flags for Segment segName changed to segNewFlags  
**Data field**  
Contains the following:  
- Time stamp  
- Segment objid  
- New segment flags  
- Old segment flags  
**Condition**  
This event is generated when the following internal flags for a segment are changed:  
REMOVED USER_ADDED LAYOUT_OFF  
USE_XY BUS_SEG STAR_SEG  
TOKEN_RING FDDI_RING SERIAL_SEG |
| 50790445 NFC_EV NodeFlagsChgd Node Flags changed | **Descr field**  
Node Flags changed to nodeFlags  
**Data field**  
Contains the following:  
- Time stamp  
- Node object ID  
- NewNodeFlags  
- OldNodeFlags  
**Condition**  
This event is generated when the following internal flags for a node change:  
REMOVED USER_ADDED LAYOUT_OFF  
USE_XY CONNECTOR GATEWAY  
STAR_HUB IS_SMART_CONN IS_BRIDGE |
### Table 33. Tivoli NetView Internal Traps (continued)

<table>
<thead>
<tr>
<th>Number</th>
<th>UNIX name</th>
<th>Windows name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>50790446</td>
<td>IFC_EV</td>
<td>IntfFlagsChgd</td>
<td>Interface Flags changed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Descr field</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Data field</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Condition</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50790466</td>
<td>CSP_EV</td>
<td>ChangeSvcPollInts</td>
<td>Change Service Polling Intervals</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Descr field</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Data field</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Condition</strong></td>
</tr>
<tr>
<td>50790467</td>
<td>SMAC_EV</td>
<td>ServmonAction</td>
<td>Servmon Action</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Descr field</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Data field</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Condition</strong></td>
</tr>
<tr>
<td>58720256</td>
<td>CPUL_EV</td>
<td>CPULoad</td>
<td>CPU Load</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Descr field</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Data field</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Condition</strong></td>
</tr>
</tbody>
</table>
Table 33. Tivoli NetView Internal Traps (continued)

<table>
<thead>
<tr>
<th>Number, UNIX name, Windows name, and Description</th>
<th>Fields and Condition</th>
</tr>
</thead>
</table>
| 58720257  
DSPU_EV  
DiskSpace%Use  
Disk Space Percentage Used | **Descr field**  
Disk space percentage used *string*  
**Data field**  
Disk usage  
**Condition**  
If the disk space usage exceeds a specified threshold, this event is generated. This event is obsolete. The SNMP Data Collector now performs all thresholding. |
| 58720258  
IPD_EV  
Intf%Deferred  
Interface Percent Deferred | **Descr field**  
Interface percent Deferred *interfaceAddr*  
**Data field**  
New value.  
**Condition**  
If the percent packets deferred exceeds a specified threshold, then this event gets generated. This is obsolete functionality. The SNMP Data Collector now performs all thresholding. |
| 58720259  
IPC_EV  
Intf%Collisns  
Interface Percent Collisions | **Descr field**  
Interface Percent Collisions *interfaceInetAddr*  
**Data field**  
New value.  
**Condition**  
If the percent collision packets exceeds a specified threshold, then this event gets generated. This is obsolete functionality. The SNMP Data Collector now performs all thresholding. |
| 58720260  
ICE_EV  
IntfCRCErrors  
Interface CRC Errors | **Descr field**  
Interface CRC Error *interfaceInetAddr*  
**Data field**  
New value.  
**Condition**  
If the number of CRC errors exceeds a specified threshold, then this event gets generated. This is obsolete functionality. The SNMP Data Collector now performs all thresholding. |
| 58720261  
IPIE_EV  
Intf%InErrs  
Interface Percent Input Errors | **Descr field**  
Interface Percent Input Error *interfaceInetAddr*  
**Data field**  
New value  
**Condition**  
If the percent input errors exceeds a specified threshold, then this event gets generated. This is obsolete functionality. The SNMP Data Collector now performs all thresholding. |
### Table 33. Tivoli NetView Internal Traps (continued)

<table>
<thead>
<tr>
<th>Number, UNIX name, Windows name, and Description</th>
<th>Fields and Condition</th>
</tr>
</thead>
</table>
| 58720262 IPOE_EV Intf%OutErrs Interface Percent Output Errors | **Descr field**: Interface Percent Output Errors `interfaceInetAddr`  
**Data field**: New value  
**Condition**: If the percent output errors exceeds a specified threshold, then this event gets generated. This is obsolete functionality. The SNMP Data Collector now performs all thresholding. |
| 58720263 DCOL_EV CollDetThresh Data Collector Detected Threshold | **Descr field**: `mibAlias instance threshold exceeded (>thresholdValue): snmpValue`  
**Data field**: `mibName`  
**Condition**: The SNMP data collector generates this event when it detects that a threshold MIB value has been exceeded. |
| 58720264 DCRA_EV CollRe-arm Data Collector re-arm event | **Descr field**: `mibAlias instance threshold rearmed (<=resetValue): snmpValue`. Sampled high of `highValue` at `highTime`  
**Data field**: `mibName`  
**Condition**: The SNMP data collector generates this event when it detects that a thresholded MIB variable has descended below the rearm value. The MIB variable is then rearmed for the threshold event DCOL_EV. |
| 58785792 IADD_EV IntfAdded Interface Added | **Descr field**: It can be one of the following strings:  
- Interface `ifLabel` Added  
- Connection Added to `segmentName`  
- Connection Added  
**Data field**: Contains the following:  
- IP address  
- Time stamp  
- Node and interface objids  
**Condition**: The netmon daemon has successfully pinged a node and detected a new interface. |
<table>
<thead>
<tr>
<th>Number, UNIX name, Windows name, and Description</th>
<th>Fields and Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>58785793 IDEL_EV IntfDeleted Interface Deleted</td>
<td><strong>Descr field</strong>&lt;br&gt;It can be one of the following strings:&lt;br&gt;- Interface ifLabel Deleted&lt;br&gt;- Connection Deleted to segmentName&lt;br&gt;- Connection Deleted&lt;br&gt;<strong>Data field</strong>&lt;br&gt;Contains the following:&lt;br&gt;- IP address&lt;br&gt;- Time stamp&lt;br&gt;- Node and interface objids&lt;br&gt;<strong>Condition</strong>&lt;br&gt;The interface is deleted in response to either a user action or as directed by netmon.</td>
</tr>
<tr>
<td>58785794 NADD_EV NodeAdded Node Added</td>
<td><strong>Descr field</strong>&lt;br&gt;Node Added&lt;br&gt;<strong>Data field</strong>&lt;br&gt;Time stamp and node objid&lt;br&gt;<strong>Condition</strong>&lt;br&gt;The node is added to the topology database.</td>
</tr>
<tr>
<td>58785795 NDEL_EV NodeDeleted Node Deleted</td>
<td><strong>Descr field</strong>&lt;br&gt;Node Deleted&lt;br&gt;<strong>Data field</strong>&lt;br&gt;Contains the following:&lt;br&gt;- Time stamp&lt;br&gt;- Node objid&lt;br&gt;<strong>Condition</strong>&lt;br&gt;The node is removed from the topology database.</td>
</tr>
<tr>
<td>58785796 HSRPADD_EV HSRPAdded Hot Standby Router Protocol (HSRP) interface added</td>
<td><strong>Descr field</strong>&lt;br&gt;HSRP interface added&lt;br&gt;<strong>Data field</strong>&lt;br&gt;Contains the following:&lt;br&gt;- Time stamp&lt;br&gt;- Node objid&lt;br&gt;<strong>Condition</strong>&lt;br&gt;An HSRP IP address was discovered by netmon.</td>
</tr>
<tr>
<td>58785797 HSRPDEL_EV HSRPDeleted Hot Standby Router Protocol (HSRP) interface deleted</td>
<td><strong>Descr field</strong>&lt;br&gt;HSRP interface deleted&lt;br&gt;<strong>Data field</strong>&lt;br&gt;Contains the following:&lt;br&gt;- Time stamp&lt;br&gt;- Node objid&lt;br&gt;<strong>Condition</strong>&lt;br&gt;Either an HSRP IP address has been taken over by another router, or is no longer being used.</td>
</tr>
</tbody>
</table>
Table 33. Tivoli NetView Internal Traps (continued)

<table>
<thead>
<tr>
<th>Number, UNIX name, Windows name, and Description</th>
<th>Fields and Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>58851329</strong> ERR_EV NonFatalError Non-Fatal errors</td>
<td>Most of these are internal debugging messages. The “Agent in distress” messages have been seen with agents that are causing trouble</td>
</tr>
<tr>
<td><strong>58851330</strong> FERR_EV FatalErrors Fatal Errors</td>
<td>It can be one of the following strings. The number here corresponds with the number under the Condition field.</td>
</tr>
</tbody>
</table>

**Descri field**

- See Table 34 on page 314 for the Descr field values.

**Data field**

- NULL

**Condition**

- See Table 34 on page 314 for the conditions.

1. Fatal ERROR Out of memory -- exiting.
2. Fatal ERROR Could not map field fieldName into OVwFieldId
3. Fatal ERROR> <Node/Iface or memory allocation error in where - exiting
4. Netmon probably stopped and ungracefully disconnected from trapd
5. snmpCollect probably stopped and ungracefully disconnected from trapd
6. topmd probably stopped and ungracefully disconnected from trapd
7. applicationName reached maximum number of outstanding events, disconnecting from trapd.
8. Application reached maximum number of outstanding events, disconnecting from trapd.

**Data field**

- NULL

**Condition**

- Following are the conditions:
  1. When allocation of dynamic memory fails.
  2. Failed to convert a list of field names into corresponding field IDs.
  3. System ran out of memory.
  4. trapd has detected that the connection to netmon has closed, but netmon did not send a close_event.
  5. trapd has detected that the connection to snmpCollect has closed, but snmpCollect did not send a close_event.
  6. trapd has detected that the connection to ovtopmd has closed, but ovtopmd did not send a close_event.
  7. trapd is closing the connection to applicationName because trapd has reached the maximum number of outstanding events that it will queue up.
  8. trapd is closing the connection to the application, because trapd has queue up the maximum number of events for the application.
Table 33. Tivoli NetView Internal Traps (continued)

<table>
<thead>
<tr>
<th>Number, UNIX name, Windows name, and Description</th>
<th>Fields and Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>58916864 NUP_EV NodeUp Node Up</td>
<td>Descr field Node Up</td>
</tr>
<tr>
<td>Data field</td>
<td>Contains the following:</td>
</tr>
<tr>
<td></td>
<td>• Time stamp</td>
</tr>
<tr>
<td></td>
<td>• Object ID</td>
</tr>
<tr>
<td>Condition</td>
<td>All detected interfaces for the node are up.</td>
</tr>
<tr>
<td>58916865 NDWN_EV NodeDown Node Down</td>
<td>Descr field Node Down</td>
</tr>
<tr>
<td>Data field</td>
<td>Contains the following:</td>
</tr>
<tr>
<td></td>
<td>• Time stamp</td>
</tr>
<tr>
<td></td>
<td>• Object ID</td>
</tr>
<tr>
<td>Condition</td>
<td>All detected interfaces for the node are down.</td>
</tr>
<tr>
<td>58916866 IUP_EV IntfUp Interface Up</td>
<td>Descr field Interface interfaceLabel up.</td>
</tr>
<tr>
<td>Data field</td>
<td>The IP address, time stamp of the event, node ID and interface ID.</td>
</tr>
<tr>
<td>Condition</td>
<td>An interface that was previously down has responded to a ping.</td>
</tr>
<tr>
<td>58916867 IDWN_EV IntfDown Interface Down</td>
<td>Descr field Interface interfaceLabel Down.</td>
</tr>
<tr>
<td>Data field</td>
<td>The IP address, time stamp of the event, node ID and interface ID.</td>
</tr>
<tr>
<td>Condition</td>
<td>An interface that was previously up is not responding to a ping.</td>
</tr>
<tr>
<td>58916868 SC_EV SegCritical Segment Critical</td>
<td>Descr field Segment segmentId Down</td>
</tr>
<tr>
<td>Data field</td>
<td>Contains the following:</td>
</tr>
<tr>
<td></td>
<td>• Time stamp</td>
</tr>
<tr>
<td></td>
<td>• Segment object ID</td>
</tr>
<tr>
<td>Condition</td>
<td>All nodes within a segment are critical.</td>
</tr>
<tr>
<td>58916869 NC_EV NetCritical Network Critical</td>
<td>Descr field Network netName Down</td>
</tr>
<tr>
<td>Data field</td>
<td>Time stamp of the event and network objid</td>
</tr>
<tr>
<td>Condition</td>
<td>All segments for the network are critical.</td>
</tr>
<tr>
<td>Number, UNIX name, Windows name, and Description</td>
<td>Fields and Condition</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>58916871 SNMP_EV SNMPStatus SNMP Status Event - Not generated</td>
<td>This is used for configurable status. The MIB variables contain: Varbind-1: 14 Varbind-2: <em>Selection name of object</em> Varbind-3: “Object status is” Varbind-4: One of the following: Unknown Normal Marginal Critical Up Down User1 User2</td>
</tr>
<tr>
<td>58916872 SUGUP_EV UNIX Only Mid-Level Manager Up</td>
<td>Descr field Mid-Level Manager <em>nodeName</em> Up Data field Time stamp of the event and the node name Condition The Mid-Level Manager is running on this node.</td>
</tr>
<tr>
<td>58916873 SUGDN_EV UNIX Only Mid-Level Manager Down</td>
<td>Descr field Mid-Level Manager <em>nodeName</em> Down Data field Time stamp of the event and the node name Condition The Mid-Level Manager is not running on this node.</td>
</tr>
<tr>
<td>58916964 NV6KUP_EV UNIX Only Tivoli NetView Up</td>
<td>Descr field Tivoli NetView <em>nodeName</em> Up Data field Contains the following: Time stamp Node name Condition Tivoli NetView is running on this node.</td>
</tr>
<tr>
<td>58916965 NV6KDN_EV UNIX Only Tivoli NetView Down</td>
<td>Descr field Tivoli NetView <em>nodeName</em> Down Data field Contains the following: Time stamp Node name Condition Tivoli NetView is no longer running on this node.</td>
</tr>
<tr>
<td>Number, UNIX name, Windows name, and Description</td>
<td>Fields and Condition</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>----------------------</td>
</tr>
</tbody>
</table>
| 58916966 IASD_EV IntfUser1 Interface Administratively Down | **Descr field** Interface `interfacename` Administrative Down  
**Data field** Timestamp of the event and objid of the node object  
**Condition** The Tivoli NetView product has detected that an interface is administratively down. |
| 58916968 NETUNREACH_EV NetworkUnreach Network Unreachable | **Descr field** Network `IPAddress` is Unreachable, mask `SubnetMask`  
**Data field** Timestamp of the event and objid of the node object  
**Condition** The Tivoli NetView product has detected that a subnet is unreachable. |
| 58916969 NETREACH_EV NetworkReach Network Reachable | **Descr field** Network `IPAddress` is Reachable again, mask `SubnetMask`  
**Data field** Contains the following:  
Timestamp of the event and objid of the node object  
Selection name  
Subnet IP Address  
Subnet mask  
**Condition** The Tivoli NetView product has detected that a subnet is reachable again. |
| 58916970 IUNREACH_EV IfaceUnreach Interface Unreachable | **Descr field** Interface `interfacename` Unreachable  
**Data field** Timestamp of the event and objid of the node object  
**Condition** The Tivoli NetView product has detected that an interface is unreachable. |
| 58916971 ROUTDOWN_EV RouterDown Router Down | **Descr field** `RouterName` Down  
**Data field** Timestamp of the event and objid of the node object  
**Condition** The Tivoli NetView product has detected that a router is down. |
<table>
<thead>
<tr>
<th>Number, UNIX name, Windows name, and Description</th>
<th>Fields and Condition</th>
</tr>
</thead>
</table>
| **58916972** ROUTUNREACH_EV RouterUnreach Router Unreachable | **Descr field** Router *RouterName* Unreachable  
**Data field** Timestamp of the event and objid of the node object  
**Condition** The Tivoli NetView product has detected that a router is unreachable. |
| **58916973** ROUTERUP_EV RouterUp Router Up | **Descr field** Router Up  
**Data field** Timestamp of the event and objid of the node object  
**Condition** The Tivoli NetView product has detected that a router is up. |
| **58916974** ROUTMARG_EV RouterMarg Router Marginal | **Descr field** Router Marginal  
**Data field** Timestamp of the event and objid of the node object  
**Condition** The Tivoli NetView product has detected that a router is in a marginal state. |
| **58916975** SUP_EV ServiceUp Service Up | **Descr field** Service *servicename* Normal  
**Data field** The service object ID, node object ID, service symbol label name, integer old status, integer new status, separated by a space.  
**Condition** The servmon daemon has detected that the status of service *servicename* has changed to Normal. |
| **58916976** SDWN_EV ServiceDown Service Down | **Descr field** Service *servicename* Critical  
**Data field** The service object ID, node object ID, service symbol label name, integer old status, integer new status, separated by a space.  
**Condition** The servmon daemon has detected that the status of service *servicename* has changed to Critical. |
| **58916977** SMAR_EV ServiceMarginal Service Marginal | **Descr field** Service *servicename* Marginal  
**Data field** The service object ID, node object ID, service symbol label name, integer old status, integer new status, separated by a space.  
**Condition** The servmon daemon has detected that the status of service *servicename* has changed to Marginal. |
<table>
<thead>
<tr>
<th>Number, UNIX name, Windows name, and Description</th>
<th>Fields and Condition</th>
</tr>
</thead>
</table>
| 58916978  
SACK_EV  
ServiceAcknowledged  
Service Acknowledged | **Descr field**  
*Service servicename* Acknowledged  
**Data field**  
None  
**Condition**  
This trap is issued by the Tivoli NetView product when a user manually acknowledges a service. The user can only manually acknowledge a service on Tivoli NetView for Windows. The servmon daemon does not respond to this trap. |
| 58916979  
SUACK_EV  
ServiceUnacknowledged  
Service Unacknowledged | **Descr field**  
*Service servicename* Unacknowledged  
**Data field**  
None  
**Condition**  
This trap is issued by the Tivoli NetView product when a user manually unacknowledges a service. The user can only manually unacknowledge a service on Tivoli NetView for Windows. The servmon daemon does not respond to this trap. |
| 58916980  
MSER_EV  
ServiceManaged  
Service Managed | **Descr field**  
*Service servicename* Managed  
**Data field**  
None  
**Condition**  
When a user manages a specific service (on Windows only) or manages a node containing the service (all platforms), this trap is issued by the Tivoli NetView product and processed by servmon. The servmon daemon resumes monitoring the named service on the named node. |
| 58916981  
USER_EV  
ServiceUnmanaged  
Service Unmanaged | **Descr field**  
*Service servicename* Unmanaged  
**Data field**  
None  
**Condition**  
When a user unmanages a specific service (on Windows only) or unmanages a node containing the service (all platforms), this trap is issued by the Tivoli NetView product and processed by servmon. The servmon daemon stops monitoring the named service on the named node until the service or node is managed again. |
Table 33. Tivoli NetView Internal Traps (continued)

<table>
<thead>
<tr>
<th>Number, UNIX name, Windows name, and Description</th>
<th>Fields and Condition</th>
</tr>
</thead>
</table>
| 58916982 IBUACTIVE_EV ISDNBackActive ISDN Backup Active | **Descr field**  
Backup line ACTIVE on interface `interfacename` (`IPaddress`)  
**Data field**  
Contains the following:  
Node name  
Formatted description of the event  
Database name  
**Condition**  
The ISDN Backup Interface is now active. |
| 58916983 IBUDORMANT_EV ISDNBackDormant ISDN Backup Dormant | **Descr field**  
Backup line DORMANT on interface `interfacename` (`IPaddress`)  
**Data field**  
Contains the following:  
Node name  
Formatted description of the event  
Database name  
**Condition**  
The ISDN Backup Interface is now dormant. |
| 58916984 L2UP_EV Layer2Up Layer 2 device is UP | **Descr field**  
Layer 2 device UP  
**Data field**  
Contains the following:  
Time stamp  
Node ID  
**Condition**  
The Tivoli NetView product has detected from the Tivoli Switch Analyzer events that the status of the switch has changed. |
| 58916985 L2DOWN_EV Layer2Down Layer 2 device DOWN | **Descr field**  
Layer 2 device DOWN  
**Data field**  
Contains the following:  
Time stamp  
Node ID  
**Condition**  
The Tivoli NetView product has detected from the Tivoli Switch Analyzer events that the status of the switch has changed. |
| 58916986 L2MARG_EV Layer2Marginal Layer 2 device is MARGINAL | **Descr field**  
Layer 2 device MARGINAL  
**Data field**  
Contains the following:  
Time stamp  
Node ID  
**Condition**  
The Tivoli NetView product has detected from the Tivoli Switch Analyzer events that the status of the switch has changed. |
Table 33. Tivoli NetView Internal Traps (continued)

<table>
<thead>
<tr>
<th>Number, UNIX name, Windows name, and Description</th>
<th>Fields and Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>58982400</td>
<td><strong>Descr field</strong></td>
</tr>
<tr>
<td>LLAC_EV</td>
<td>Link Address For <code>interfaceLabel</code> Changed to <code>physicalAddress</code></td>
</tr>
<tr>
<td>LLAddressChgd</td>
<td><strong>Data field</strong></td>
</tr>
<tr>
<td></td>
<td>The IP address, time stamp of the event, node and interface object IDs</td>
</tr>
<tr>
<td></td>
<td><strong>Condition</strong></td>
</tr>
<tr>
<td></td>
<td>The Tivoli NetView product has detected that <code>interfaceLabel</code> has a new physical Address.</td>
</tr>
</tbody>
</table>

<p>| 58982401                                        | <strong>Descr field</strong>       |
| MLLA_EV                                         | <code>nodeName1</code> reported different Link Address than obtained from <code>nodeName2</code> by SNMP |
| MismatchLLAddr                                  | <strong>Data field</strong>        |
|                                                | <code>nodeName</code> of the node that reported different address |
|                                                | <strong>Condition</strong>         |
|                                                | A discrepancy in the link level address (LLA) being reported by two nodes had been detected. The ifTable is being interrogated for the LLA and has been compared to the information in the <code>nodeName2</code> ARP cache. Use the <code>rnetstat -I nodeName1</code> command to determine the interfaces and LLA for <code>nodeName1</code>, and then issue the <code>rnetstat -A nodeName2 | grep nodeName1</code> command to determine the <code>nodeName2</code> ARP entry for <code>nodeName1</code>. |
|                                                | It is possible that the information stored in the topology database for <code>nodeName1</code> is out of date. To resolve this problem, perform a demand poll on <code>nodeName1</code>. If this does not resolve the problem, issue the <code>ovtopodump -L | grep nodeName1</code> command to determine the LLA that is currently associated with the interfaces for <code>nodeName1</code>. |
|                                                | <strong>Note:</strong> Some vendors make wide assumptions about documenting information in the ARP cache, thus there are many reasons why a mismatch can occur, but the network configuration is still valid. Netmon supports a flag to disable this check completely. To disable, select the option <code>ignore</code> for Ring bit-swapping storage flag through the Tivoli desktop Set Options for netmon daemon job. If you are not using the Tivoli desktop, refer to the netmon man page for information about setting the argument for ignoring ring bit-swapping. |</p>
<table>
<thead>
<tr>
<th>Number, UNIX name, Windows name, and Description</th>
<th>Fields and Condition</th>
</tr>
</thead>
</table>
| 58982402 ULLA_EV UndtrmdLLAddr Undetermined Link Level Address | Descr field This trap can contain one of the following description fields:  
- If there are two nodes that have reported different LLA addresses, the following message is displayed: “nodeName reports a different Link Address for this node from that reported by secondNodeName”  
- If the same node reports a different LLA address than what is reported earlier, the following message is displayed “nodeName reports a different Link Address for this node than it reported earlier”.  
Data field  
nodeName, secondNodeName  
nodeName The name of the node that reported a different LLA address than the one that this (reporting) node received earlier either from the same node (nodeName = secondNodeName) or from a different node (nodeName = secondNodeName)  
secondNodeName The name of the node that which had earlier reported the LLA address.  
Condition  
nodeName provides an LLA address that is different than the one provided by the IP address. If this message keeps repeating, it indicates that the LLA for the interface is repeatedly going up and down; this usually means that there is more than one interface with the same IP address. If it happens once or twice and then stops, it usually means that a LAN card has been changed.  
There might be valid network reasons why the ULLA_EV is being displayed. To disable this message, perform a demand poll on nodeName1. If this does not resolve the problem, issue the ovtopodump -L | grep nodeName1 command to determine the LLA that is currently associated with the interfaces for nodeName1. |
| 58982403 OIC EV ObjectIdChg Object ID Change | Descr field Object Identifier changed to newObjectId  
Data field Time stamp, node object ID, new sysObject ID  
Condition A new sysObjectID was detected |
<table>
<thead>
<tr>
<th>Number, UNIX name, Windows name, and Description</th>
<th>Fields and Condition</th>
</tr>
</thead>
</table>
| 58982404  
SDC_EV  
SysDescrChg  
System Description Change | **Descr field**  
System Description Changed to sysDescr  
**Data field**  
Contains the following:  
- Time stamp  
- Node object ID  
- New sysDescr  
**Condition**  
A new sysDescr was detected |
| 58982405  
SNC_EV  
SysNameChg  
System Name Change | **Descr field**  
System Name changed  
**Data field**  
Contains the following:  
- Time stamp  
- Node object ID  
- New sysName  
**Condition**  
A new sysName was detected |
| 58982406  
SMC_EV  
SubnetMaskChg  
Subnet Mask Change | **Descr field**  
Network Mask for interfaceLabel changed to inetAddr  
**Data field**  
The IP address, time stamp of the event, node ID, interface id, and interface type.  
**Condition**  
A new subnet mask for an interface was detected |
| 58982407  
FSC_EV  
ForwdStatChg  
Forwarding Status Change | **Descr field**  
It can be one of the following fields:  
- Forwarding status changed — Now a Gateway  
- Forwarding status changed — Now a host  
- Forwarding status changed — Now Unknown  
**Data field**  
Contains the following:  
- Time stamp  
- Node object ID  
- 1 if it is a gateway and 2 if it is a host  
**Condition**  
A change was detected in the ipForwarding status for the node. |
<table>
<thead>
<tr>
<th>Number, UNIX name, Windows name, and Description</th>
<th>Fields and Condition</th>
</tr>
</thead>
</table>
| 58982408  
FTH_EV  
ForwardToHost  
Forwarding to a host | **Descr field**  
Incorrect Routing to node `nodeName`  
**Data field**  
`nodeName`  
**Condition**  
The node for the event contains a route in its routing table to node `nodeName`, but the topology database entry for `nodeName` indicates that its `ipForwarding` status is not a gateway. To correct this problem use a demand poll to node `nodeName` to update its `ipForwarding` status in the database. This event could also be generated if node `nodeName` is not allowing SNMP communication to the manager. In this case, the manager cannot determine the `ipForwarding` status for `nodeName`. |
| 58982410  
SCC_EV  
SysContactChg  
System Contact Change | **Descr field**  
System Contact Changed to `contactName`  
**Data field**  
Contains the following:  
• Time stamp  
• Node object ID  
• New system contact  
**Condition**  
The system contact for the node has changed. |
| 58982411  
SLC_EV  
SysLocChg  
System Location Change | **Descr field**  
System Location Changed to `nodeLocation`  
**Data field**  
Contains the following:  
• Time stamp  
• Node object ID  
• New system location  
**Condition**  
The system location for the node has changed. |
| 58982412  
ITC_EV  
IntfTypeChg  
Interface Type Change | **Descr field**  
Interface Type for `interfaceLabel` changed to `typeName`  
**Data field**  
IP address, time stamp, node id, interface ids and interface type.  
**Condition**  
The interface type has changed. |
| 58982413  
IDC_EV  
IntfDescChg  
Interface Description Change | **Descr field**  
Interface Descriptor for `interfaceLabel` changed to `interfaceName`  
**Data field**  
IP address, time stamp, node id, interface ids and interface description.  
**Condition**  
The interface description has changed. |
<table>
<thead>
<tr>
<th>Number, UNIX name, Windows name, and Description</th>
<th>Fields and Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>58982414 BSM_EV BadSubnetMask Failed Subnet Mask</td>
<td>Descri field: Inconsistent subnet mask subnetMask on interface inetAddr</td>
</tr>
<tr>
<td></td>
<td>Data field: The interface address and subnet mask of the interface</td>
</tr>
<tr>
<td></td>
<td>Condition: The subnet mask obtained for an IP address does not match the assumed subnet mask for the subnet in which that interface resides.</td>
</tr>
<tr>
<td>58982415 DUPIP_EV DupNodeName Duplicate IP</td>
<td>Descri field: Duplicate IP address IPAddress on nodeName and nodeName</td>
</tr>
<tr>
<td></td>
<td>Data field: IP address</td>
</tr>
<tr>
<td></td>
<td>Condition: A duplicate IP address has been found.</td>
</tr>
<tr>
<td>58982416 NORM_EV NormalizeObject Normalized</td>
<td>Descri field: It can be one of the following strings:</td>
</tr>
<tr>
<td></td>
<td>- Normalized interface IPAddress detected, where IPAddress is a dotted decimal IP address</td>
</tr>
<tr>
<td></td>
<td>- Normalized node nodeSelectionName detected, where nodeSelectionName is node’s OVwDb selection name</td>
</tr>
<tr>
<td></td>
<td>- Normalized interface IPAddress is no longer normalized, where IPAddress is a dotted decimal IP address</td>
</tr>
<tr>
<td></td>
<td>- Normalized node nodeSelectionName is no longer normalized, where nodeSelectionName is a node’s OVwDb selection name</td>
</tr>
<tr>
<td></td>
<td>Data field: IP address</td>
</tr>
<tr>
<td></td>
<td>Condition: The Tivoli NetView product has detected an interface or node with an IP address that has normalized or is no longer normalized.</td>
</tr>
<tr>
<td>58982417 SADD_EV ServiceAdded Service Added</td>
<td>Descri field: Service servicename added</td>
</tr>
<tr>
<td></td>
<td>Data field: The service object ID and node object ID separated by a space</td>
</tr>
<tr>
<td></td>
<td>Condition: The servicename has been discovered by the servmon daemon.</td>
</tr>
<tr>
<td>58982418 SDEL_EV ServiceDeleted Service Deleted</td>
<td>Descri field: Service servicename deleted</td>
</tr>
<tr>
<td></td>
<td>Data field: The service object ID and node object ID separated by a space</td>
</tr>
<tr>
<td></td>
<td>Condition: This event is generated by the servmon daemon when it detects that service servicename has been deleted from a node.</td>
</tr>
<tr>
<td>Number, UNIX name, Windows name, and Description</td>
<td>Fields and Condition</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>58982419 SNMPADDR_EV SNMPAddress SNMP Address</td>
<td>Descr field SNMP address changed to ipaddress</td>
</tr>
<tr>
<td></td>
<td>Data field Contains the following: Node name IP address</td>
</tr>
<tr>
<td></td>
<td>Condition This is the IP address that the Tivoli NetView product is using for the SNMP polling.</td>
</tr>
<tr>
<td>58982420 MPLSENABLED_EV MPLSEnabled MPLS Enabled</td>
<td>Descr field MPLS support enabled</td>
</tr>
<tr>
<td></td>
<td>Data field Contains the following: Node name</td>
</tr>
<tr>
<td></td>
<td>Condition MPLS has been enabled on this router.</td>
</tr>
<tr>
<td>58982421 MPLSDISABLED_EV MPLSDisabled MPLS Disabled</td>
<td>Descr field MPLS support disabled</td>
</tr>
<tr>
<td></td>
<td>Data field Contains the following: Node name</td>
</tr>
<tr>
<td></td>
<td>Condition MPLS has been disabled on this router.</td>
</tr>
<tr>
<td>58982422 ATADD_EV ServiceAttributeAdded Service Attribute Added</td>
<td>Descr field Service Attribute \textit{attributename} Added</td>
</tr>
<tr>
<td></td>
<td>Data field The node object ID, a blank space, and the field ID of the capability field that was added to the node</td>
</tr>
<tr>
<td></td>
<td>Condition This trap is generated when a node object is assigned a new attribute (Boolean capability field), but a corresponding service object is not created. This occurs when an entry in the /usr/OV/conf/servmon.conf file does not specify a value for the service SmartSet name and service SmartSet label fields.</td>
</tr>
<tr>
<td>58982423 ATDEL_EV ServiceAttributeRemoved Service Attribute Deleted</td>
<td>Descr field Service Attribute \textit{attributename} Deleted</td>
</tr>
<tr>
<td></td>
<td>Data field The node object ID, a blank space, and the field ID of the capability field that was removed from the node</td>
</tr>
<tr>
<td></td>
<td>Condition This trap is generated when the attribute (Boolean capability field) that was added by the servmon daemon is deleted. Note that this trap is not generated when an attribute is removed from a node object when a service object is deleted. The length of time a service can remain Critical before it is deleted is specified by the Deletion\textunderscore Interval value in the /usr/OV/conf/service\textunderscore polling.conf file.</td>
</tr>
<tr>
<td>Number, UNIX name, Windows name, and Description</td>
<td>Fields and Condition</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>----------------------</td>
</tr>
</tbody>
</table>
| 58982424 MVSSYSTEM_EV MvsSystemChg Node Configuration Events | **Descr field** MVS system name changed to *systemname*  
**Data field** Time stamp, node objid, and new MVS system name  
**Condition** The MVS system name has changed. |
| 58982425 MVSSYSPLEX_EV MvsSysplexChg Node Configuration Events | **Descr field** MVS sysplex name changed to *sysplexname*  
**Data field** Time stamp, node objid, and new MVS sysplex name  
**Condition** The MVS sysplex name has changed. |
| 58982426 MVSPROC_EV MvsProcChg Node Configuration Events | **Descr field** MVS procedure name changed to *procname*  
**Data field** Time stamp, node objid, and new MVS procedure name  
**Condition** The MVS procedure name has changed. |
| 59047936 AA_EV GenAppAlert Application Alert | **Descr field** alertapp: classStr: alertMsg  
**alertApp** The application name which has sent the alert  
**classStr** Can take one of the following values: INFORMATION WARNING ERROR DISASTER  
**alertMsg** The Alert message sent by the application  
**Data field** NULL  
**Condition** An application used the OVwAlertMsg API to send a message. |
| 59179056 APUP_EV AppUp Application Up event | **Descr field** applicationName connected to trapd  
**Data field** NULL  
**Condition** An application has connected to the trapd daemon using its internal socket API. |
### Table 33. Tivoli NetView Internal Traps (continued)

<table>
<thead>
<tr>
<th>Number, UNIX name, Windows name, and Description</th>
<th>Fields and Condition</th>
</tr>
</thead>
</table>
| **59179057**  
APDN_EV  
AppDown  
Application Down Event | **Descr field**  
applicationName disconnecting from trapd  
**Data field**  
NULL  
**Condition**  
An application has disconnected from the trapd daemon. |
| **59179068**  
TATM_EV  
TraceMaskChg  
Tralertd Change Tracemask Event | **Descr field**  
Tralertd trace mask change request  
**Data field**  
tracemask  
**Condition**  
The tralertd daemon uses this to change its tracemask. |
| **59179070**  
NMCR_EV  
NMRetryCntChg  
Netmon Change Retry Count Event | **Descr field**  
Netmon retry count changed  
**Data field**  
retry count value  
**Condition**  
This event is generated when the netmon option to change the retry count is issued. This event is obsolete with the retries allowed using the ovsnmp.conf file. |
| **59179071**  
ACTF_EV  
UNIX Only  
Action Failure | **Descr field**  
temporary script [line number] : error message  
**Data field**  
Contains only zeros — no additional data is provided  
**Condition**  
An error was detected when the commands were run by the ESE actionsvr daemon. |
| **59179072**  
NVCOLD_UP  
UNIX Only  
nvcold Initialization Event | **Descr field**  
nvcold initialization complete  
**Data field**  
NULL  
**Condition**  
This event is generated when the nvcold daemon completes its initialization phase. |
<table>
<thead>
<tr>
<th>Number, UNIX name, Windows name, and Description</th>
<th>Fields and Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>59179073 DISPSUB dispsub Display Submap</td>
<td><strong>Descr field</strong> Requested object: <em>hostname</em> (IPAddress). Contains the following: <strong>Varbind-1</strong>: Name <strong>Varbind-2</strong>: Optional IP address <strong>Varbind-3</strong>: NetView map name <strong>Varbind-4</strong>: NetView x-display name <strong>Varbind-5</strong>: Optional commands which are used when the Application type = 1: 0 ping 2 nmdemandpoll 4 netcheck 6 rnetstat 8 findroute <strong>Varbind-6</strong>: Application type: 0 xnmbrowser 1 xnmappmon 99 Display submap and highlight node <strong>Varbind-7</strong>: Optional node name to be used with the command. <strong>Condition</strong> This event causes a Tivoli NetView submap to be displayed.</td>
</tr>
<tr>
<td>1879048240 NVOT001 UNIX Only Vertex Created</td>
<td><strong>Descr field</strong> Can be one of the following values: • namebinding(int) • protocol(int) • name(string) • detailsld(string) <strong>Condition</strong> A vertex in gtmd has been created.</td>
</tr>
<tr>
<td>1879048241 NVOT002 UNIX Only Vertex Deleted</td>
<td><strong>Descr field</strong> One of the following values: • namebinding(int) • protocol(int) • name(string) <strong>Condition</strong> A vertex has been deleted in gtmd.</td>
</tr>
<tr>
<td>1879048242 NVOT003 UNIX Only Vertex Status Changed</td>
<td><strong>Descr field</strong> Can be one of the following values: • namebinding(int) • protocol(int) • name(string) <strong>Condition</strong> The status of a vertex in gtmd has changed.</td>
</tr>
</tbody>
</table>
### Table 33. Tivoli NetView Internal Traps (continued)

<table>
<thead>
<tr>
<th>Number, UNIX name, Windows name, and Description</th>
<th>Fields and Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1879048243 NVOT004 UNIX Only Graph Created</td>
<td>Descr field Can be one of the following values:</td>
</tr>
<tr>
<td></td>
<td>• namebinding(int)</td>
</tr>
<tr>
<td></td>
<td>• protocol(int)</td>
</tr>
<tr>
<td></td>
<td>• name(string)</td>
</tr>
<tr>
<td></td>
<td>• detailsld(string)</td>
</tr>
<tr>
<td></td>
<td>Condition A graph or box has been created in gtmd.</td>
</tr>
<tr>
<td>1879048244 NVOT005 UNIX Only Graph Deleted</td>
<td>Descr field Can be one of the following values:</td>
</tr>
<tr>
<td></td>
<td>• namebinding(int)</td>
</tr>
<tr>
<td></td>
<td>• protocol(int)</td>
</tr>
<tr>
<td></td>
<td>• name(string)</td>
</tr>
<tr>
<td></td>
<td>Condition A graph or box has been deleted from gtmd.</td>
</tr>
<tr>
<td>187904825 NVOT006 UNIX Only Arc Created</td>
<td>Descr field Can be one of the following values:</td>
</tr>
<tr>
<td></td>
<td>• namebinding(int)</td>
</tr>
<tr>
<td></td>
<td>• Aprotocol(int)</td>
</tr>
<tr>
<td></td>
<td>• Aname(string)</td>
</tr>
<tr>
<td></td>
<td>• Zprotocol(int/str)</td>
</tr>
<tr>
<td></td>
<td>• Zname(string)</td>
</tr>
<tr>
<td></td>
<td>• arcindexld(int)</td>
</tr>
<tr>
<td></td>
<td>• detailsld(string)</td>
</tr>
<tr>
<td></td>
<td>Condition An arc has been created in gtmd.</td>
</tr>
<tr>
<td>1879048246 NVOT007 UNIX Only Arc Deleted</td>
<td>Descr field Can be one of the following values:</td>
</tr>
<tr>
<td></td>
<td>• namebinding(int)</td>
</tr>
<tr>
<td></td>
<td>• Aprotocol(int)</td>
</tr>
<tr>
<td></td>
<td>• Aname(string)</td>
</tr>
<tr>
<td></td>
<td>• Zprotocol(int/str)</td>
</tr>
<tr>
<td></td>
<td>• Zname(string)</td>
</tr>
<tr>
<td></td>
<td>• arcindexld(int)</td>
</tr>
<tr>
<td></td>
<td>Condition An arc has been created in gtmd.</td>
</tr>
<tr>
<td>1879048247 NVOT008 UNIX Only Arc Status Changed</td>
<td>Descr field Can be one of the following values:</td>
</tr>
<tr>
<td></td>
<td>• namebinding(int)</td>
</tr>
<tr>
<td></td>
<td>• Aprotocol(int)</td>
</tr>
<tr>
<td></td>
<td>• Aname(string)</td>
</tr>
<tr>
<td></td>
<td>• Zprotocol(int/str)</td>
</tr>
<tr>
<td></td>
<td>• Zname(string)</td>
</tr>
<tr>
<td></td>
<td>• arcindexld(int)</td>
</tr>
<tr>
<td></td>
<td>Condition The status of an arc in gtmd has changed.</td>
</tr>
<tr>
<td>Number, UNIX name, Windows name, and Description</td>
<td>Fields and Condition</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>77777770 CNATP_Started CNATP_Started Primary CNAT started</td>
<td><strong>Descr field</strong> None <strong>Data field</strong> None <strong>Condition</strong> This trap indicates that the primary CNAT has been started either by the CNAT administrator using the CNAT Configuration and Control Console, or when the primary CNAT system has been restarted and it has been configured to start automatically when it is restarted. This trap does not indicate that the primary CNAT has started performing translation.</td>
</tr>
<tr>
<td>77777771 CNATP_Stopped CNATP_Stopped Primary CNAT stopped</td>
<td><strong>Descr field</strong> None <strong>Data field</strong> None <strong>Condition</strong> This trap indicates that the primary CNAT has been stopped either by the CNAT administrator using the CNAT Configuration and Control Console, or when the primary CNAT system has been restarted.</td>
</tr>
<tr>
<td>77777772 CNATP_StartXlate CNATP_StartXlate Primary CNAT started translating</td>
<td><strong>Descr field</strong> None <strong>Data field</strong> None <strong>Condition</strong> This trap indicates that the primary CNAT has started translating. This trap is issued by the primary CNAT when it starts translating IP traffic, either because the CNAT administrator started the primary CNAT using the CNAT Configuration and Control Console, or because the primary CNAT has resumed translating after a network connection has been restored.</td>
</tr>
<tr>
<td>77777773 CNATP_StopXlate CNATP_StopXlate Primary CNAT stopped translating</td>
<td><strong>Descr field</strong> None <strong>Data field</strong> None <strong>Condition</strong> This trap indicates that the primary CNAT has stopped translating, either because the CNAT administrator stopped the primary CNAT using the CNAT Configuration and Control Console, or because the primary CNAT detected a network failure to the adjacent router.</td>
</tr>
<tr>
<td>Number, UNIX name, Windows name, and Description</td>
<td>Fields and Condition</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td><strong>7777774</strong> CNATR_Started CNATR_Started Redundant CNAT started</td>
<td><strong>Descr field</strong> None</td>
</tr>
<tr>
<td></td>
<td><strong>Data field</strong> None</td>
</tr>
<tr>
<td></td>
<td><strong>Condition</strong> This trap indicates that the redundant CNAT has been started either by the CNAT administrator using the CNAT Configuration and Control Console, or when the redundant CNAT system has been restarted and it has been configured to start automatically when it is restarted. This trap does not indicate that the redundant CNAT has started performing translation.</td>
</tr>
<tr>
<td><strong>7777775</strong> CNATR_Stopped CNATR_Stopped Redundant CNAT stopped</td>
<td><strong>Descr field</strong> None</td>
</tr>
<tr>
<td></td>
<td><strong>Data field</strong> None</td>
</tr>
<tr>
<td></td>
<td><strong>Condition</strong> This trap indicates that the redundant CNAT has been stopped, either by the CNAT administrator using the CNAT Configuration and Control Console, or because the redundant CNAT system has been restarted.</td>
</tr>
<tr>
<td><strong>7777776</strong> CNATR_StartXlate CNATR_StartXlate Redundant CNAT started translating</td>
<td><strong>Descr field</strong> None</td>
</tr>
<tr>
<td></td>
<td><strong>Data field</strong> None</td>
</tr>
<tr>
<td></td>
<td><strong>Condition</strong> This trap indicates that the redundant CNAT has started translating. This trap is issued by the redundant CNAT when it starts translating IP traffic, either because the CNAT administrator started the redundant CNAT using the CNAT Configuration and Control Console, or because the redundant CNAT has started translating instead of the primary CNAT, because the primary CNAT cannot be reached.</td>
</tr>
<tr>
<td><strong>7777777</strong> CNATR_StopXlate CNATR_StopXlate Redundant CNAT stopped translating</td>
<td><strong>Descr field</strong> None</td>
</tr>
<tr>
<td></td>
<td><strong>Data field</strong> None</td>
</tr>
<tr>
<td></td>
<td><strong>Condition</strong> This trap indicates that the redundant CNAT has stopped translating. This trap is issued by the redundant CNAT when it stops translating IP traffic, either because the CNAT administrator stopped the redundant CNAT using the CNAT Configuration and Control Console, or because the redundant CNAT has started translating instead of the primary CNAT, because the redundant CNAT detected that the primary CNAT has resumed translating.</td>
</tr>
<tr>
<td>Descr Field</td>
<td>Condition</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
</tr>
<tr>
<td>1. ERROR Internal Error - object of bad type <code>objectType</code> detected in addObj</td>
<td>1. The program enters in default case in the function <code>addObjToTopo()</code>.</td>
</tr>
<tr>
<td>2. ERROR Internal Error - object of bad type <code>objectType</code> detected in remove</td>
<td>2. The program enters in default case in the function <code>removeObjFromTopo()</code>.</td>
</tr>
<tr>
<td>3. ERROR Internal Error - object of bad type <code>objectType</code> detected in delete</td>
<td>3. The program enters in default case in the function <code>deleteObjFromTopo()</code>.</td>
</tr>
<tr>
<td>5. ERROR Could not create main database directory: <code>errorNumber</code></td>
<td>5. Creation of <code>/usr/OV/databases/&lt;dbname&gt;</code> fails.</td>
</tr>
<tr>
<td>6. ERROR Could not allocate object ID for interface <code>ifName</code></td>
<td>6. Creation of object ID for an interface fails.</td>
</tr>
<tr>
<td>7. ERROR Could not allocate object ID for net <code>netName</code></td>
<td>7. Creation of object ID for a network fails.</td>
</tr>
<tr>
<td>8. ERROR Could not allocate Topo Object ID for Node <code>nodeName</code></td>
<td>8. Creation of object ID for a node fails.</td>
</tr>
<tr>
<td>9. ERROR Unable to format time in format <code>formatDBtime</code>, return value = <code>retVal</code></td>
<td>9. Conversion of time format between system and INGRES/SQL formats failed.</td>
</tr>
<tr>
<td>10. ERROR Failed to get the field id from selection name.</td>
<td>10. Failed to get the field id from selection name.</td>
</tr>
<tr>
<td>11. ERROR Failed to get unique object name from selection name ID and base name.</td>
<td>11. Failed to get unique object name from selection name ID and base name.</td>
</tr>
<tr>
<td>12. Either failed to create object id or failed to get an object ID for an already existing object.</td>
<td>12. Either failed to create object id or failed to get an object ID for an already existing object.</td>
</tr>
<tr>
<td>13. An attempt is made to increase the number of fields in the bind list when it is already full.</td>
<td>13. An attempt is made to increase the number of fields in the bind list when it is already full.</td>
</tr>
<tr>
<td>14. Unable to set the value of the fields in the bind list.</td>
<td>14. Unable to set the value of the fields in the bind list.</td>
</tr>
<tr>
<td>15. The number of maximum topology specific fields which an object can have is <code>MAX_UNSET_FIDS</code>. During cleanup if more than 50 fields are found this event is sent.</td>
<td>15. The number of maximum topology specific fields which an object can have is <code>MAX_UNSET_FIDS</code>. During cleanup if more than 50 fields are found this event is sent.</td>
</tr>
<tr>
<td>16. Failed to convert a field name to field ID.</td>
<td>16. Failed to convert a field name to field ID.</td>
</tr>
<tr>
<td>17. When an attempt is made to add another value to a field which is of list type and already the field has maximum number of permissible values.</td>
<td>17. When an attempt is made to add another value to a field which is of list type and already the field has maximum number of permissible values.</td>
</tr>
<tr>
<td>18. Failed to get field ID from field name.</td>
<td>18. Failed to get field ID from field name.</td>
</tr>
<tr>
<td>19. Failed to get field ID from field name.</td>
<td>19. Failed to get field ID from field name.</td>
</tr>
<tr>
<td>20. Failed to get field ID from field name.</td>
<td>20. Failed to remove network from topology.</td>
</tr>
<tr>
<td>21. Failed to get field ID from field name.</td>
<td>21. Failed to remove segment from topology.</td>
</tr>
<tr>
<td>22. Failed to get field ID from field name.</td>
<td>22. 'Stat' call fails on pipe.</td>
</tr>
<tr>
<td>23. A pipe descriptor was expected but it is not a pipe.</td>
<td>23. A pipe descriptor was expected but it is not a pipe.</td>
</tr>
<tr>
<td>24. Failed to create a pipe.</td>
<td>24. Failed to create a pipe.</td>
</tr>
<tr>
<td>25. Failed to open a pipe (fdopen call failed).</td>
<td>25. Failed to open a pipe (fdopen call failed).</td>
</tr>
<tr>
<td>26. Core dump has occurred while running netmon.</td>
<td>26. Core dump has occurred while running netmon.</td>
</tr>
<tr>
<td>27. Fatal error has occurred.</td>
<td>27. Fatal error has occurred.</td>
</tr>
<tr>
<td>28. The SNMP wait queue has nodes to process, yet the node contains no state information.</td>
<td>28. The SNMP wait queue has nodes to process, yet the node contains no state information.</td>
</tr>
<tr>
<td>29. The specified node is spinning in its ipRouteTable. Aborting requests.</td>
<td>29. The specified node is spinning in its ipRouteTable. Aborting requests.</td>
</tr>
<tr>
<td>30. The specified node is spinning in its ifTable. Aborting requests to the node.</td>
<td>30. The specified node is spinning in its ifTable. Aborting requests to the node.</td>
</tr>
<tr>
<td>31. The specified node is spinning in its ipAddrTable. Aborting requests.</td>
<td>31. The specified node is spinning in its ipAddrTable. Aborting requests.</td>
</tr>
</tbody>
</table>
Appendix B. Tivoli NetView Events for the Tivoli Enterprise Console

This section provides information about the events that are forwarded by the Tivoli NetView product to the Tivoli Enterprise Console product. Table 35 provides information about the events that are forwarded by default. Table 36 on page 316 provides information about the other events that can be forwarded.

Event Mapping and New Class Structure

Table 35 lists the events that are forwarded by the Tivoli NetView product to the Tivoli Enterprise Console product by default.

<table>
<thead>
<tr>
<th>ClassName</th>
<th>Replaces</th>
<th>Event</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEC_ITS_INTERFACE_ADDED</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interface deleted</td>
<td>OV_If_Deleted</td>
<td>58785793 (IDEL_EV)</td>
<td>DELETED</td>
</tr>
<tr>
<td>TEC_ITS_INTERFACE_MANAGE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interface unmanaged</td>
<td>OV_Unmanage_IF</td>
<td>50790442 (UI_EV)</td>
<td>UNMANAGE</td>
</tr>
<tr>
<td>TEC_ITS_INTERFACE_STATUS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interface Up</td>
<td>OV_IF_Down</td>
<td>58916866 (IUP_EV)</td>
<td>UP</td>
</tr>
<tr>
<td>Interface Down</td>
<td>NEW</td>
<td>58916867 (IDWN_EV)</td>
<td>DOWN</td>
</tr>
<tr>
<td>Interface Admin Down</td>
<td>NEW</td>
<td>58916966 (IASD_EV)</td>
<td>AdminDown</td>
</tr>
<tr>
<td>Interface Unreachable</td>
<td></td>
<td>58916970 (IUNREACH_EV)</td>
<td>UNREACHABLE</td>
</tr>
<tr>
<td>TEC_ITS_ISDN_STATUS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISDN Active</td>
<td>NEW</td>
<td>58916982 (IBUACTIVE_EV)</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>ISDN Dormant</td>
<td>NEW</td>
<td>58916983 (IBUDORMANT_EV)</td>
<td>DORMANT</td>
</tr>
<tr>
<td>TEC_ITS_NODE_ADDED</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node deleted</td>
<td>OV_Node_Deleted</td>
<td>58785795 (NDEL_EV)</td>
<td>DELETED</td>
</tr>
<tr>
<td>TEC_ITS_NODE_MANAGE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node unmanaged</td>
<td>OV_Unmanage_Node</td>
<td>50790419 (UN_EV)</td>
<td>UNMANAGE</td>
</tr>
<tr>
<td>TEC_ITS_NODE_STATUS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node Up</td>
<td>OV_Node_Down</td>
<td>58916864 (NUP_EV)</td>
<td>UP</td>
</tr>
<tr>
<td>Node Down</td>
<td>OV_Node_Down</td>
<td>58916865 (NDWN_EV)</td>
<td>DOWN</td>
</tr>
<tr>
<td>Node Marginal</td>
<td>OV_Node_Marginal</td>
<td>50790400 (NM_EV)</td>
<td>MARGINAL</td>
</tr>
</tbody>
</table>
### Table 35. Events Forwarded by Default (continued)

<table>
<thead>
<tr>
<th>ClassName</th>
<th>Replaces</th>
<th>Event</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEC_ITS_ROUTER_STATUS</td>
<td>NV_Router_Status</td>
<td>58916971</td>
<td>ROUTDOWN_EV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>58916972</td>
<td>ROUTUNREACH_EV</td>
</tr>
<tr>
<td>Router Down</td>
<td></td>
<td>58916973</td>
<td>ROUTERUP_EV</td>
</tr>
<tr>
<td>Router Unreachable</td>
<td></td>
<td>58916974</td>
<td>ROUTMARG_EV</td>
</tr>
<tr>
<td>Router Up</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Router Marginal</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>58916971</td>
<td>ROUTDOWN_EV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>58916972</td>
<td>ROUTUNREACH_EV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>58916973</td>
<td>ROUTERUP_EV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>58916974</td>
<td>ROUTMARG_EV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_SNMP_COLLECT_THRESHOLD</td>
<td>OV_DataCollectThresh</td>
<td>58720263</td>
<td>DCOL_EV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>58720264</td>
<td>DCRA_EV</td>
</tr>
<tr>
<td>SNMP Collect threshold exceeded</td>
<td>NV_Subnet_Reachability</td>
<td>58916968</td>
<td>NETUNREACH_EV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>58916969</td>
<td>NETREACH_EV</td>
</tr>
</tbody>
</table>

**Note:** TEC_ITS_SA_STATUS events are also forwarded by default. See documentation for the optional IBM Tivoli Switch Analyzer product for more information.

Table 36 lists the event mapping and new class structure for events that are not listed in Table 35 on page 315.

### Mappings between Events and New Class Structure

### Table 36. Events and New Class Structure

<table>
<thead>
<tr>
<th>Class Name</th>
<th>Replaces</th>
<th>Event</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEC_ITS_APPLICATION_ALERT</td>
<td>NV6k_Application_Alert</td>
<td>59047936</td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_APPLICATION_STATUS</td>
<td>NV6k_Application_Up_Event</td>
<td>5919056</td>
<td>UP</td>
</tr>
<tr>
<td></td>
<td>NV6k_Application_Down_Event</td>
<td>5919057</td>
<td>DOWN</td>
</tr>
<tr>
<td>TEC_ITS ASN MIB DEF FILE_FORMAT</td>
<td>NV6k ASN MIB Def_File_Format</td>
<td>50790439</td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_BAD_SUBNET_MASK</td>
<td>OV_Bad_Subnet_Mask</td>
<td>58982414</td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_CHANGE_IF_SEGMENT</td>
<td>OV_Chg_If_Segment</td>
<td>50790427</td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_CHANGE_NETMON_RETRY_COUNT</td>
<td>NV6k Change_NetmonRetry_Count</td>
<td>59179070</td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_CHANGE_POLLING_PERIOD</td>
<td>OV_Change_Polling_Period</td>
<td>50790411</td>
<td></td>
</tr>
<tr>
<td>TEC_ITS CMIS EVENT</td>
<td>OV_CMIS_Event</td>
<td>58916870</td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_CONNECTION_ADDED</td>
<td>OV_Connection_Added</td>
<td>50790409</td>
<td>ADDED</td>
</tr>
<tr>
<td>(Windows Only)</td>
<td>OV_Connection_Deleted</td>
<td>50790410</td>
<td>DELETED</td>
</tr>
<tr>
<td>TEC_ITS_CPU_LOAD</td>
<td>NV6k_Cpu_Load</td>
<td>58720256</td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_DISK_SPACE</td>
<td>NV6k Disk_Space_Percentage_Used</td>
<td>58720257</td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_ERROR</td>
<td>OV_Error</td>
<td>58851329</td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_FATAL_ERROR</td>
<td>OV_Fatal_Error</td>
<td>58851330</td>
<td></td>
</tr>
</tbody>
</table>
### Table 36. Events and New Class Structure (continued)

<table>
<thead>
<tr>
<th>Class Name</th>
<th>Replaces</th>
<th>Event</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEC_ITS_FORCED POLL</td>
<td>NV6K_Forced_Poll</td>
<td>50790412</td>
<td>START</td>
</tr>
<tr>
<td></td>
<td>NV6K_Cancel_Forced_Poll</td>
<td>50790413</td>
<td>CANCELLED</td>
</tr>
<tr>
<td>TEC_ITS_FORW_STATUS CHG</td>
<td>OV_Forw_Status_Chg</td>
<td>58982407</td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_FORWARDING TO A HOST</td>
<td>NV6K_Forwarding_To_A_Host</td>
<td>58982408</td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_HSRP_ADDED</td>
<td>NV6K_Hsrp_Added</td>
<td>58785796</td>
<td>ADDED</td>
</tr>
<tr>
<td>(UNIX Only)</td>
<td>NV6K_Hsrp_Deleted</td>
<td>58785797</td>
<td>DELETED</td>
</tr>
<tr>
<td>TEC_ITS_IF_CRC_ERROR</td>
<td>NV6K_Interface_Crc_Errors</td>
<td>58720260</td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_IF_DESC_CHANGED</td>
<td>OV_If_Desc_Chg</td>
<td>58982413</td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_IF_FLAGS_CHANGED</td>
<td>OV_If_Flags_Chg</td>
<td>50790446</td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_IF_PERCENT_DEFERRED</td>
<td>NV6K_Interface_Percent_Defered</td>
<td>58720258</td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_IF_PERCENT_COLLISIONS</td>
<td>NV6K_Interface_Percent_Collisions</td>
<td>58720259</td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_IF_PERCENT_INPUT_ERRORS</td>
<td>NV6K_Interface_Percent_Input_Errors</td>
<td>58720261</td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_IF_PERCENT_OUTPUT_ERRORS</td>
<td>NV6K_INTERFACE_PERCENT_OUTPUT_ERRORS</td>
<td>58720262</td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_IF_POSITION_CHANGED</td>
<td>NV6K_Interface_Position_Changed</td>
<td>50790447</td>
<td></td>
</tr>
<tr>
<td>(Windows Only)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_IF_TYPE_CHANGED</td>
<td>OV_If_Type_Changed</td>
<td>58982412</td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_INTERFACE_ADDED</td>
<td>OV_If_Added</td>
<td>58785792</td>
<td>ADDED</td>
</tr>
<tr>
<td>TEC_ITS_INTERFACE_MANAGE</td>
<td>OV_Manage_If</td>
<td>50790441</td>
<td>MANAGE</td>
</tr>
<tr>
<td>TEC_ITS_INTERFACE_MANAGE</td>
<td>OV_Manage_Network</td>
<td>50790442</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_L2_NODE_STATUS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Layer 2 device is UP</td>
<td>NEW</td>
<td>58916984</td>
<td>UP EV</td>
</tr>
<tr>
<td></td>
<td>Layer 2 device is DOWN</td>
<td>NEW</td>
<td>DOWN EV</td>
</tr>
<tr>
<td></td>
<td>Layer 2 device is MARGINAL</td>
<td>NEW</td>
<td>MARGINAL EV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_MAP_CHANGE</td>
<td>NV6K_Map_Change_Event</td>
<td>59113474</td>
<td></td>
</tr>
<tr>
<td>(Windows Only)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_NETMON_ACTION</td>
<td>NV6K_Netmon_Action</td>
<td>50790426</td>
<td></td>
</tr>
<tr>
<td>(Windows Only)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_NETMON_CHANGE_TRACE_FILE</td>
<td>NV6K_Netmon_Change_Trace_File</td>
<td>50790422</td>
<td></td>
</tr>
<tr>
<td>(Windows Only)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_NETMON_CHANGE_TRACE_MASK</td>
<td>NV6K_Netmon_Change_Trace_Mask</td>
<td>50790423</td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_NETMON_DUMP_NODE</td>
<td>NV6K_Netmon_Dump_Node</td>
<td>50790424</td>
<td></td>
</tr>
<tr>
<td>(Windows Only)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_NETMON_DUMP_INTERFACE</td>
<td>NV6K_Netmon_Dump_Interface</td>
<td>50790425</td>
<td></td>
</tr>
<tr>
<td>(Windows Only)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_NETWORK_ADDED</td>
<td>OV_Network_Added</td>
<td>50790407</td>
<td>ADDED</td>
</tr>
<tr>
<td></td>
<td>OV_Network_Deleted</td>
<td>50790408</td>
<td>DELETED</td>
</tr>
<tr>
<td>TEC_ITS_NETWORK_FLAGS_CHANGED</td>
<td>OV_Network_Flg_Chg</td>
<td>50790443</td>
<td></td>
</tr>
<tr>
<td>TEC_ITS_NETWORK_MANAGE</td>
<td>OV_Manage_Network</td>
<td>50790416</td>
<td>MANAGE</td>
</tr>
<tr>
<td></td>
<td>OV_Unmanage_Network</td>
<td>50790417</td>
<td>UNMANAGE</td>
</tr>
</tbody>
</table>

Appendix B. Tivoli NetView Events for the Tivoli Enterprise Console 317
<table>
<thead>
<tr>
<th>Class Name</th>
<th>Replaces</th>
<th>Event</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TEC_ITS_NETWORK_STATUS</strong></td>
<td>OV_Network_Normal</td>
<td>50790403</td>
<td>UP</td>
</tr>
<tr>
<td></td>
<td>OV_Network_Marginal</td>
<td>50790404</td>
<td>MARGINAL</td>
</tr>
<tr>
<td></td>
<td>OV_Network_Critical</td>
<td>58976869</td>
<td>DOWN</td>
</tr>
<tr>
<td><strong>TEC_ITS_NODE_ADDED</strong></td>
<td>OV_Node_Added</td>
<td>58785794</td>
<td>ADDED</td>
</tr>
<tr>
<td><strong>TEC_ITS_NODE_FLAGS_CHANGED</strong></td>
<td>OV_Node_Flags_Chg</td>
<td>50790445</td>
<td></td>
</tr>
<tr>
<td><strong>TEC_ITS_NODE_MANAGE</strong></td>
<td>OV_Manage_Node</td>
<td>50790418</td>
<td>MANAGE</td>
</tr>
<tr>
<td><strong>TEC_ITS_NO_SNMP_REPLY</strong></td>
<td>OV_No_Snmp_Reply</td>
<td>58982409</td>
<td></td>
</tr>
<tr>
<td><strong>TEC_ITS_OBJECT_ID_CHANGE</strong></td>
<td>OV_Object_Id_Chg</td>
<td>58982403</td>
<td></td>
</tr>
<tr>
<td><strong>TEC_ITS_PHYS_ADDR_CHANGED</strong></td>
<td>OV_Phy_Addr_Chg</td>
<td>58982400</td>
<td></td>
</tr>
<tr>
<td><strong>TEC_ITS_PHYS_ADDR_MISMATCH</strong></td>
<td>OV_Phy_Addr_Mismatch</td>
<td>58982401</td>
<td></td>
</tr>
<tr>
<td><strong>TEC_ITS_SEGMENT_ADDED</strong></td>
<td>OV_Segment_added</td>
<td>50790405</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OV_Segment_Deleted</td>
<td>50790406</td>
<td></td>
</tr>
<tr>
<td><strong>TEC_ITS_SEGMENT_FLAGS_CHANGED</strong></td>
<td>OV_Segment_Flag_Chg</td>
<td>50790444</td>
<td></td>
</tr>
<tr>
<td><strong>TEC_ITS_SEGMENT_MANAGE</strong></td>
<td>OV_Manage_Segment</td>
<td>50790420</td>
<td>MANAGE</td>
</tr>
<tr>
<td></td>
<td>OV_Unmanage_Segment</td>
<td>50790421</td>
<td>UNMANAGE</td>
</tr>
<tr>
<td><strong>TEC_ITS_SEGMENT_STATUS</strong></td>
<td>OV_Segment_Marginal</td>
<td>50790401</td>
<td>UP</td>
</tr>
<tr>
<td></td>
<td>OV_Segment_Marginal</td>
<td>50790402</td>
<td>MARGINAL</td>
</tr>
<tr>
<td></td>
<td>OV_Segment_Critical</td>
<td>58916868</td>
<td>DOWN</td>
</tr>
<tr>
<td><strong>TEC_ITS_SERVER_STATUS</strong></td>
<td>NV6k_Up</td>
<td>58916964</td>
<td>UP</td>
</tr>
<tr>
<td></td>
<td>NV6k_Down</td>
<td>58916965</td>
<td>DOWN</td>
</tr>
<tr>
<td><strong>TEC_ITS_SERVICE_POINT_APPL_CHANGED_MASK (Windows Only)</strong></td>
<td>NV6k_Service_Point_Appl_Changed_Mask</td>
<td>59179069</td>
<td></td>
</tr>
<tr>
<td><strong>TEC_ITS_SERVICE_STATUS</strong></td>
<td>NV_Service_Up</td>
<td>58916975</td>
<td>UP</td>
</tr>
<tr>
<td></td>
<td>NV_Service_Down</td>
<td>58916976</td>
<td>DOWN</td>
</tr>
<tr>
<td><strong>TEC_ITS_SNMP_DATA_COL_FILE_FORMAT</strong></td>
<td>NV6k_Sys_Data_Col_File_Format</td>
<td>50790440</td>
<td></td>
</tr>
<tr>
<td><strong>TEC_ITS_SNMP_STATUS_CHANGE</strong></td>
<td>NV6k_Snmp_Status_Change</td>
<td>58916871</td>
<td></td>
</tr>
<tr>
<td><strong>TEC_ITS_SUBNET_MASK_CHANGED</strong></td>
<td>OV_Subnet_Mask_Chg</td>
<td>58982406</td>
<td></td>
</tr>
<tr>
<td><strong>TEC_ITS_SYS_CONTACT_CHANGED</strong></td>
<td>OV_Sys_Contact_Chg</td>
<td>58982410</td>
<td></td>
</tr>
<tr>
<td><strong>TEC_ITS_SYS_DESCR_CHANGED</strong></td>
<td>OV_Sys_Desc_Chng</td>
<td>58982404</td>
<td></td>
</tr>
<tr>
<td><strong>TEC_ITS_SYS_LOCATION_CHANGED</strong></td>
<td>OV_Sys_Location_Chg</td>
<td>58982411</td>
<td></td>
</tr>
<tr>
<td><strong>TEC_ITS_SYS_NAME_CHANGED</strong></td>
<td>OV_Sys_Name_Chg</td>
<td>58982405</td>
<td></td>
</tr>
<tr>
<td><strong>TEC_ITS_TRAPLET_CHANGE_TRACEMASK</strong></td>
<td>NV6k_Traplet_Change_Tracemask_Event</td>
<td>59179068</td>
<td></td>
</tr>
<tr>
<td><strong>TEC_ITS_TRAP_CONF_FORMAT_CHANGED</strong></td>
<td>NV6k_TRAPD.CONF_Format_Changed</td>
<td>50790438</td>
<td></td>
</tr>
<tr>
<td><strong>TEC_ITS_THRESHOLD_CHANGE</strong></td>
<td>NV6k_Set_Threshold</td>
<td>50790414</td>
<td>SET</td>
</tr>
<tr>
<td></td>
<td>NV6k_Delete_Threshold</td>
<td>50790415</td>
<td>DELETE</td>
</tr>
<tr>
<td><strong>TEC_ITS_UNDETERMINED_LINK_LEVEL_ADDR</strong></td>
<td>NV6k_Undetermined_Link_Level_Addr</td>
<td>58982402</td>
<td></td>
</tr>
<tr>
<td><strong>TEC_ITS_WARNING</strong></td>
<td>NV6k_Warnings</td>
<td>50462720</td>
<td></td>
</tr>
</tbody>
</table>
Appendix C. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.
IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM’s future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to
IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

If you are viewing this information in softcopy form, the photographs and color illustrations might not appear.

**Trademarks**

AIX, DB2, IBM, the IBM logo, IBMLink, MVS, OS/2, Tivoli, the Tivoli logo, Tivoli Enterprise Console, and TME are trademarks or registered trademarks of International Business Machines Corporation or Tivoli Systems Inc. in the United States, other countries, or both.

Intel, the Intel Inside logos, MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft and Windows NT are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.
Appendix D. Additional Copyright and License Information

The product described in this document also contains software downloaded from several Web servers. Permission to download and use such software is conditional upon inclusion of the following notices.

gd 1.2 © Copyright 1994, 1995, Quest Protein Database Center, Cold Spring Harbor Labs. Permission granted to copy and distribute this work provided that this notice remains intact. Credit for the library must be given to the Quest Protein Database Center, Cold Spring Harbor Labs, in all derived works. This does not affect your ownership of the derived work itself, and the intent is to assure proper credit for Quest, not to interfere with your use of gd. If you have questions, ask. (“Derived works” includes all programs that utilize the library. Credit must be given in user-visible documentation.)

gd 1.2 was written by Thomas Boutell and is currently distributed by boutell.com, Inc.

If you wish to release modifications to gd, please clear them first by sending email to boutell@boutell.com; if this is not done, any modified version of the gd library must be clearly labeled as such.

The Quest Protein Database Center is funded under Grant P41-RR02188 by the National Institutes of Health.

Written by Thomas Boutell, 2/94–8/95.

The GIF compression code is based on that found in the pbmplus utilities, which in turn is based on GIFENCOD by David Rowley. See the notice below:

Based on GIFENCOD by David Rowley. A Lemple-Ziv compression based on “compress”.

Modified by Marcel Wijkstra.

Copyright © 1989 by Jef Poskanzer.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission appear in all supporting documentation. This software is provided “as is” without express or implied warranty.

The Graphics Interchange Format © is the Copyright property of CompuServe Incorporated. GIF (sm) is a Service Mark property of CompuServe Incorporated.

The GIF decompression is based on that found in the pbmplus utilities, which in turn is based on GIFDECOD by David Koblas. See the notice below:


Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above
copyright notice appears in all copies and that both that copyright notice and this permission appear in all supporting documentation. This software is provided “as is” without express or implied warranty.

GIFtrans v1.12

Convert any GIF file into a GIF89a. Allows for setting the transparent or background color, changing colors, adding or removing comments. Also code to analyze GIF contents.

Copyright © 24.2.94 by Andreas Ley

Permission to use, copy, modify, and distribute this software for any purpose and without fee is hereby granted, provided that the above copyright notice appears in all copies. This software is provided “as is” without express or implied warranties.
Index

Special characters
Xdefaults file  99

A
access levels to maps  56
accessibility information  x
acknowledging, unacknowledging objects
  global-based mode
    from a map  82
    from an event  83
    overview  82
    status as an object attribute  82
    using nvdbimport  83
import/export utility  83
map-based mode  82
overview  81
action ruleset node  124
actionsvr daemon  13
activating event correlation rulesets  138
activating, deactivating event filters  148
addalert command  160
adding
  and changing background
    graphics  95, 96
    connections  70
  correlation ruleset node  137
  manager-container associations  188
  objects and symbols  68
  SmartSets  86
addtrap command  157
algorithms, layout
  automatic layout, setting  97
  changing  78
  description  60
  redo layout, setting  96
aliases, Agent Policy Manager
  description  274
  managing  274
APM (Agent Policy Manager)
  aliases
    description  274
    managing  274
  (continued)
  definition
    (continued)
    distribution failures  230
    file monitoring, dialog box
      fields  239
    file monitoring, procedure  222
    how to define and distribute
      modifying  231
    successful distribution
      230
    threshold dialog fields,
      description  237
    threshold, procedure  235
    viewing, active  229
    viewing, pending  232
    description  219
    distributing changes  220
    managing agents  219
    diagnosing problems  264
    distribution status indicators  227
    MLM domains
      description  274
      rearranging, automatically  275
      rearranging, manually  275
    requirements  219
SmartSet icons
  APM monitors  220
  MLM managers  220
  propagating status  221
  SmartSets  220
  app-defaults
    changing  100
    event card or list behavior  120
    manager backup  191
  application defaults
    changing  100
    event card or list behavior  120
    manager backup  191
  application plane  61
  application, service monitor  176
  applications
    collmap  83
    defined  64
    ipmap  2, 64
    nevents  3
    ovw  2
    performance  197
    xnmappmon  6
    xnmbrowser  4
    xnmbuilder  5
    xnmcollect  5
    xnmfault  5
    xnmgraph  5
    xnmloadmib  4
    xnmrunreport  5
    xnmssmmpconf  4
    xnmtrap  4
    xxmap  3
  ARFs to SRFs, converting  42
  ARP cache information  175
  assigning map access levels  56
  assigning submaps
    home  80
    parent  80
  audit data, security
    creating reports  40
    defining categories  34
    description  25
  authentication, network
    description  21
    login
      considerations  22
    description  22
    user profiles
      creating  29
      deleting  31
    valid characters for names and
      passwords  31
    viewing  31
  automatic layout  61, 97
  automatic network discovery  163
  information retrieved from
    nodes  163
  turning off  164
  using seed files, defining management
    regions  165

B
background  72
graphics  95
  adding, changing  95
  deleting  96
  plane  61
  processes
    actionsvr  13
    C5d  11
    gtdmd  10
    mgragentd  8
    netmon  8
    noniplotopd  9
    nvcold  9
    nvcoord  13
    nvlockd  9
    nvold  11
    nvpagerd  13
    nvsed  15
    nvsectltd  15
    nvserverd  14
    orsd  12
    otmd  10
    ovelmd  13
    ovesmd  13
    ovspmd  7
    ovtopmd  9
    ovwwdb  9
    pmld  12
    snmpCollect  14
    spappld  16
    tralerd  16
    trapid  13
    trapgend  14

325
backup manager
configuration 184
default actions, changing 191
determining current backup
time 191
default actions, changing 191
sessions 191
dialog box 187
responding to
manager down notification 190
manager restored notification 191
using a seed file to configure 185
backupd daemon 185
boolean logic 85
browsing MIBs 195
building MIB applications 201

C
c_arf2srf command 42
C5d daemon 11
starting
from the command line 225
through the Tivoli desktop 225
cache, ARP 175
cache, ovwdb 168
changing
Agent Policy Manager definition 231
backup manager default actions 191
configuration 183
host name, local maps 50
host name, NFS-mounted maps 50
control desk startup 101
event card or list behavior 120
graphical interface defaults 100
map permissions 56, 98
root permission for security administration 27
status source 154
characteristics
manager-container 185
submap 58
check route ruleset node 125
checking application messages 105
checking configured interfaces 175
child submap
creating with default settings 78
creating with modified settings 78
default settings 78
steps for creating with default settings 78
steps for creating with modified settings 79
Cisco routers, discovering 167
client/server environment
managing maps 49
maps, local 50
deleting invalid maps 50
removing maps 51
updating host name 50
maps, NFS mounted 49
changing the server 50
improving network performance 49
collecting MIB data 194
collmap application 83
colors, changing in graphical interface 100

combining event correlation rulesets 138
container-manager associations (continued)
configuring 185
deleting 189
manage-unmanage rules 190
containment realm 66
contents, report directory 216
continuous network monitoring
description 23
shift-in and shift-out 23
control desk
preventing startup 101
starting
as icon 100
with events minimized 100
with network view 100
controlling event card or list behavior 120
convenience routines 57
converting
ARFs to SRFs 42
events to alerts 159
copying
group profile, security 33
symbols
procedure 74
rules 74
correlating events
description 122
how events are processed 122
nodes
action 124
check route 125
event attributes 126
forward 126
override 126
pager 127
pass on match 127
query database field 129
paging utility, configuring 123
query database SmartSet 128
query global variable 131
set database field 130
set global variable 131
set MIB variable 131
thresholds 132
trap settings 134
rulesets
activating 138
adding a node 137
combining rulesets 138
deleting a connection 138
deleting a node 138
event attributes 134
examples 139
inserting a ruleset 138
ruleset editor 136
sample 135
saving 138
testing 139

counter MIB values, displaying 106, 215
creating an Agent Policy Manager definition
from an existing definition 228
overview 221
Creating an Agent Policy Manager definition (continued)

Creating event filters 142
Creating filters
  Compound 146
  Trap-to-alert 151
Creating independent submaps 79
Creating MIB variable expressions 208
Creating SmartSets
  Adding 86
  Description 83
  Examples 92
  Listing objects 92
  Modifying 91
  Types 84
  Using pattern matching 85
Creating submaps
  A child, default settings 78
  A child, modified settings 78
  Child, independent 79
Creating table entries
  Activating and deactivating 152
  Creating 152
  Modifying 152
  Sorting 153
Customer support
  See software support ix
Customizing
  .Xdefaults file 99
  Event card or list display 120
  Failing resource display 101
  Grapher 102
  Graphical interface 95
  Manager backup default actions 191
  Maps 55
  Menu bar and tool palette 98
  MIB graph applications 102
  Objects and connections 68
  Submaps 182
  Symbol placement 67
  Traps 154
Cutting
  Procedure 74
  Restrictions 72
  Rules 72
Cutting, copying, pasting objects and symbols 72
dagons (continued)
  Nsververd 14
  Orsd 12
  Otmd 10
  Ovelmd 13
  Ovesmd 13
  Osvpm 7
  Ovtopmd 9
  Owvdb 9
  Pmd 12
  Snmpcollect 14
  Snmpdx 19
  Spapld 16
  Traler 16
  Trap 13
  Trapgend 14
Daemons, resolving problems with FFDC 16
Data collector, MIB 204
Databases
  General topology 18
  IP topology 18
  Map 18
  Description 17
  Information 48
  Object 17
  Object registration service 18
  Default compound status 46
  Default settings
    Child submap 78
    Graphical interface 100
  Defining a security policy
    Global settings 34
    Group profiles
      Adding 31
      Copying 33
      Creating 31
      Deleting 34
    Viewing permissions 34
    Overview 26
    User profiles
      Creating 29
      Deleting 31
      Viewing 31
  Defining SmartSets
    Adding 86
    Description 83
    Examples 92
    Listing objects 92
    Modifying 91
    Types 84
  Defining symbol characteristics 44
Deleting
  Agent Policy Manager definition 232
  Correlation ruleset connection 138
  Correlation ruleset node 138
  Group profiles, security 27
  Invalid maps on a client 50
  Menu items 99
  Objects 71
  Symbols 71
  User profiles, security 31
  Description, object, modifying 81
  Designating nodes as managers 187
  Determining manager backup sessions 191
Diagnosing problems, Agent Policy Manager definitions 264
Directory, report 216
Disability Information x
Discovering Cisco Routers 164
Discovering services on nodes 176
Discovering the network 163
Cisco routers 167
Configuration information retrieved during 163
Open topology 166
Using Openmon 166
Ways to 180
Discovering z/OS systems 165
Discovery and database daemons 7
Disk space, monitoring 199
Display presentation 106
Displaying multiple protocols 67
Displaying, hiding grid 106
Displaying, owvdb object database MIB interface information 180
MIB system information 180
Distributing
  Agent Policy Manager definition distribution failures 230
  Procedure 229
  Status indicators 227
  Successful distribution 230
  Security configuration 38
Distribution status indicators, Agent Policy Manager 227
Dynamic increase, file system and paging space 211
Dynamic workspace, creating 119
E
Editing rules 67
Enabling the Agent Policy Manager daemon 225
Enterprise-specific MIBs
  Browsing 195
  Loading and unloading 193
Environment variables, security 34
Event adapter, controlling 113
Event and trap processing daemons 11
Event attributes ruleset node 126
Event history 121
Event log changing
  Color of cards and text 100
  Number of events 100
  Presentation 100
  Size 121, 122
  Viewing 121
Events
  Card or list behavior, controlling 120
  Configuring 153
  Converting to alerts 159
  Correlating, rulesets 122
  Activating 138
  Adding a node 137
  Combining rulesets 138
  Deleting a connection 138
  Deleting a node 138
  Environment variables for trap data 135

Index 327
file monitoring Agent Policy Manager

definition (continued)

creating from an existing definition 228
deleting 232
diagnosing problems 264
dialog box fields, description 239
distributing

distribution failures 230
procedure 229
successful distribution 230
how to define and distribute 221
modifying 231
procedure 222
viewing
active 229
pending 232
filter editor 143
filtered command 121
filtering events 148
activating 149
deactivating 150
filters
activating and deactivating 148
creating 142
customizing for users 102
types 142
forwarding to the Tivoli Enterprise Console 110
event adapter, controlling 113
format 112
from unmanaged nodes, suppressing 120
how events are processed 122
log file 109
nodes, ruleset
action 124
check route 125
event attributes 126
forward 126
override 126
pager 127
pass on match 127
query database field 129
query database SmartSet 128
query global variable 131
set database field 130
set global variable 131
set MIB variable 131
thresholds 132
trap settings 134
paging utility, configuring 123
searching
by criteria 121
by filter 121
using addtrap command to configure 157
using mib2trap command to configure 157
viewing event log 121
examples
collecting MIB data 207
event correlation rulesets 139
internet subnet 67
partitioned internet subnet 75
SmartSets 92
using the Agent Policy Manager 269

graphs
adding lines 107, 215
changing display presentation 106
changing line configuration 104
checking application messages 105
customizing 102
displaying MIB counter values 106, 215
displaying, hiding grid 106
entering numeric values 103
generating performance reports 215
getting application statistics 105
paging, graphs 106
printing 214
scaling y-axis 106
setting time intervals 103
using context menus 107, 215
groups, security
adding 31
changing 31
copying 33
creating 31
deleting 34
description 24
pre-configured, description 24
viewing permissions 34
gtmd daemon 10

H
hide objects and symbols 71, 81
home submap, assigning 80
host connection daemons 16
host name
updating
local maps 50
NFS-mounted maps 50

I
IBM Tivoli Monitoring

defining servers to monitor 118
enabling 117
identification, network
description 21
login
considerations 22
description 22
user profiles
creating 29
deleting 31
passwords, valid characters 31
viewing 31
import/export utility for Acknowledged status 83
improving performance
imap 167
when maps are NFS-mounted 49
increasing filesystem size and paging space automatically 211
increasing ovwdb cache size 168
independent submaps
creating 79
opening 79
information in map database 48
information, accessibility x
information, disability x
interfaces
  checking configuration 175
displaying MIB information 180
internet submap
  example 67
  objects you can add 69
IP topology database 18
ipmap application
  description 2, 64
  improving performance 167
  propagating status 46
  segment topologies 66
  submaps, defining 67
  using internet submaps 66
  using network submaps 66
  using node submaps 67
  using segment submaps 66

K
  keyboard, shortcut keys x

L
  labels, modifying and displaying 97
  layout
    algorithms 60
    automatic 61, 97
  limitations during map synchronization 65
  listing
    configured services 176
    hidden objects 71, 81
    objects in a SmartSet 92
  loading MIBs 194
  local filesystem, monitoring 209
  local maps (client)
    considerations 50
    deleting invalid maps 50
    removing 51
    updating host name 50
  locating routes, nodes 175
  log file for events, changing the size 122
  logging into Tivoli NetView
    considerations 22
    description 22

M
  management regions 67
  management, process
    nvstatus 7
    ovstart 7
    ovstatus 7
    ovstop 7
  manager
    backup 184
      default actions, changing 191
      determining current backup sessions 191
      manage-unmanage rules 190
      manager down notification 190
      manager restored notification 191
      container associations
        adding 188
        manager (continued)
        container associations (continued)
        characteristics 185
        configuring 185
        deleting 189
    managing
      agents, using the Agent Policy Manager
        description 219
        requirements 219
      backup sessions
        determining current backup sessions 191
        manager down, responding 190
        manager restored, responding 191
      local maps
        deleting invalid maps 50
        removing maps 51
        updating host name 50
        maps, client/server environment 49
        network configuration 163
        network resources 193
        NFS mounted maps 49
        changing the server 50
        improving network performance 49
      SmartSets
        adding 86
        examples 92
        listing objects 92
        modifying 91
        managing and unmanaging objects 81
      SmartSets 84
      map
        applications 1
        assigning access 56, 98
        basics 48
        database 17
        permissions, setting 56, 98
        reasons for creating 55
        reasons for customizing 55
        synchronization limitations 64
        map-based Acknowledged status 82
        mapadmin command 50
        maps and submaps
          authorization 56
          client/server environment
            description 49
            local maps, considerations 50
            local maps, invalid, deleting 50
            local maps, removing a client 51
            local maps, updating host name 50
            NFS-mounted maps, changing server 50
            NFS-mounted maps
              considerations 49
              NFS-mounted maps, improving
              network performance 49
              customizing 55, 95
              database information
                editing 70
                learning about 48
              map layout 57, 60
              defining 60
              using algorithms 60
              maps and submaps (continued)
              map layout (continued)
                using automatic layout 61
                using new object holding area 61
                metacollections 61, 62
                open map 48
              snapshots 57
        menu bar
          customizing
            OvwRegDir environment variable 98
            security services 25
          messages, sending 38
        metacollection submaps
          behavior 63
          characteristics 63
          defining 62
          description 62
        migragent daemon 8
        MIB
          application builder 201
          application, adding 202
          applications, comparison 197
          browser 195
          CPU Performance 199
          data collection 203
          data, collecting 207
          disk space 199
          displaying description 196
          displaying interface information 180
          displaying system information 180
          Ethernet errors 199
          Ethernet packet types 199
          Ethernet performance 199
          Ethernet traffic 199
          graphing counter values 106, 180, 215
          interface traffic 199
          IP errors 105
          loading and unloading 193
          querying 197
          real-time performance data 198
          retrieving configuration information 180
          SNMP authentication failures 201
          SNMP errors 200
          SNMP network activity 193
          SNMP operations 200
          SNMP traffic 200
          TCP connections 200
          values, displaying 180
          variable expressions, creating and graphing 208
          mib2trap command 157
          mibExpr.conf file 208
          mibisa daemon 19
          MLM domains, Agent Policy Manager
            description 274
            rearranging
              automatically 275
              manually 275
          MLM managers map icon 220
          modem configuration file 123
          modifying
            object descriptions 81
            SmartSets 91
            submap settings 80

Index 329
modifying (continued)  
symbol labels, displaying 97  
monitoring local filesystem and paging space  
for specific events 212  
starting  
with a command 212  
with dynamic increase 211  
without dynamic increase 209  
stopping 211  
monitoring network configuration 174  
monitoring SmartSet status  
starting 84  
stopping 84

N

navigation tree, starting as icon 100  
NetView, sending alerts to 160  
network configuration  
information retrieved 163  
listing IP addresses 174  
monitoring 174  
network discovery  
automatic 163  
using openmon 166  
using seed files 165  
network monitoring, continuous  
description 23  
shift-in and shift-out 23  


network subnetmap, objects you can add 69  
network topology events 109  
network, distributed  
managing maps 49  
maps, local 50  
deleting invalid maps 50  
removing maps 51  
updating host name 50  
maps, NFS mounted 49  
changing the server 50  
improving network performance 49  
newsgroups  
NFS-mounted maps  
changing the server 50  
considerations 49  
improving network performance 49  
node configuration events 109  
nodes, ruleset  
action 124  
clear route 125  
event attributes 126  
forwarding 126  


P

pagetopology network configuration events 23  
node 69  
network 69  
root 69  
segment 69  
steps 70  


parent  
objects, definition 57  
submaps, assigning 80  
partitioned internet submap  
example 75  
steps for creating 75  
partitioned segment submap, steps for creating 77  


pattern matching 85  
performance  
gathering information 193  
improving, ipmap 167
performance (continued)  
reports, generating 215  
permissions  
map 98  
root for security administration, changing 27  
user 56  
planes  
application 61  
background 61  
user 61  
polling  
setting intervals 181  
turning on and off 169  
printing graphed data 158, 214  
process management 7  
processes  
background  
event and trap 11  
host connection 16  
ovspmd daemon 7  
process management 7  
topology and database 7  
foreground 1  
propagate status to Agent Policy Manager  
map icons 221  
propagate-at-threshold-value component status 47  
propagate-most-critical component status 47  
protocol switching 67

Q

query database field ruleset node 129  
query database SmartSet ruleset node 128  
query global variable ruleset node 131  
querying logged in users 38  
quick zoom 60

R

read-write permission 98  
redo layout 96  
remote manager status, determining 190  
removing  
invalid maps on a client 50  
maps before deinstalling a client 51  
report directory, contents 216  
report writing 217  
reports, generating and customizing on the object database  
vdbimport utility 18  
vdt format utility 18  
reports, generating performance 215  
requirements for using the Agent Policy Manager 219  
resource display, failing 101  
resources  
monitoring 209  
Xdefault 101  
retrieving MIB configuration information 180  
role of Solaris SEA agents 19  
root permission, changing for security administration 27  
root submap 58  
objects you can add 69  
routers  
fault isolation description 172  
routers, discovering Cisco 167  
Routers, Discovering Cisco 164  
routines  
ovwchgrp 57  
ovwchmod 57  
ovwchown 57  
ovwls 57  
routing tables, viewing information 175  
ruleset editor 136  
rulesets, event correlation activating 138  
creating  
adding a node 137  
combining rulesets 138  
deleting a connection 138  
deleting a node 138  
environment variables for trap data 135  
event attribute values 134  
inserting a ruleset 138  
ruleset editor 136  
description 122  
examples 139  
how events are processed 122  
nodes  
action 124  
check route 125  
event attributes 126  
forward 126  
override 126  
pager 127  
pass on match 127  
query database field 129  
query database SmartSet 128  
query global variable 131  
set database field 130  
set global variable 131  
set MIB variable 131  
thresholds 132  
trap settings 134  
paging utility, configuring 123  
sample 135  
saving 138  
testing 139  
rundap command 203

S

saving  
event correlation rulesets 138  
performance data 213  
scaling submaps 59  
scaling, y-axis 106  
SEA agents, role 19  
searching for events  
by criteria 121  
by filter 121  
security  
ARFs to SRFs, converting 42  
audit data  
creating reports 40  
security (continued)  
audit data (continued)  
defining categories 34  
description 25  
distribution, distributing 38  
converting ARFs to SRFs 42  
description 21  
distributing configuration 38  
environment variables 34  
features  
audit management 25  
consistent security controls 25  
continuous, auditable network management 23  
log in considerations 22  
log in process 22  
network access control 24  
network authentication and identification 21  
pager service 26  
password protection 23  
shift-in, shift-out operation 23  
tailored Tivoli NetView graphical interface 25  
forcing off logged in users 38  
groups  
adding 31  
changing 31  
copying 33  
creating 31  
deleting 34  
description 24  
pre-configured, description 24  
viewing permissions 34  
permissions, verifying for shell scripts 42  
querying logged in users 38  
security administration dialog  
accessing 27  
description 28  
root permission, changing 27  
sending messages 38  
security administration dialog  
accessing 34  
description 28  
root permission, changing 27  
security policy, defining  
global settings 34  


Index 331
segment submap, objects you can add 69
selectfilter command 153
selection rule, adding and changing 203
sending alerts to NetView 160
sending SNMP traps 110
service monitor
configuration file
description 178
statements 179
disabling servmon daemon 177
enabling servmon daemon 177
scheduling service discovery 178
services on nodes, discovering 176
services, listing configuration 176
servmon
custom plug-in modules for service discovery and status checking 180
set database field ruleset node 130
set global variable ruleset node 131
set MIB variable ruleset node 131
setthresh command 209
setting
compound status 47
environment variables, security 34
map permissions 98
polling intervals 181
shadow, turning off 68
shell scripts, verifying security permission 42
shift-in and shift-out 23
shortcut keys, keyboard x
shpmon application
configuring 211
dynamic increase 211
monitoring specific events 212
starting from the command line 212
simple filters, creating 144
size of graphical interface windows, changing 100
SmartSet editor
adding a SmartSet 86
elements 92
listing objects 92
modifying a SmartSet 91
using pattern matching 85
SmartSet facility 83, 84
SmartSet icons
APM monitors 220
MLM managers 220
SmartSets 220
SmartSets
maintaining 9
SmartSets, defining
adding 86
description 83
examples 92
listing objects 92
managing and unmanaging 84
modifying 91
status monitoring
starting 84
stopping 84
types 84
SmartSets, Defining and Managing 83
SmartSets, Managing and Unmanaging 84
SmartSets, Types of 84
snapshots, taking 57
SNMP nodes, configuring 181
snmpCol.conf file 5
snmpColDump command 204
snmpCollect daemon 14
snmpd daemon 19
software support
contacting ix
Solaris SEA agents, role 19
sapppld daemon 16
starting
Agent Policy Manager configuration interface 226
Agent Policy Manager daemon from the command line 225
through the Tivoli desktop 225
event display application 118
graph application 213
startup configuration file 7
status polling switch check button 169
status scheme, compound 46
submaps
assigning a parent 80
automating partitioning of 77
basics 57
characteristics 58
creating an independent 59, 79
customizing 67
home 58, 80
internet 66
metaconnections 62
modifying settings 80
network 66
partitioning 74
presentation 59
root 58
suppressing events from unmanaged nodes 120
symbols
basics 43
behavior 44
characteristics
behavior 44
label 44
location 44
status 44
type 44
variety 44
class 44
configuring creation time and buffer 167
cutting, copying, and pasting 72
deleting 71
displaying labels 97
executable 44
explodable 44
hiding 71, 81
label 44, 46
learning about 43
location 44
modifying labels 97
propagation 46
status 45
compound status source 46
symbols (continued)
status (continued)
object status source 46
symbol status source 46
type, classes and subclasses 44
variety 44
synchronization
limitations during 65
message 64
system performance
monitoring
local file system 209
paging space 209
starting with dynamic increase 211
starting without dynamic increase 211
T

tables
configuration information, nodes 163
customizing 190
default compound status scheme 46
map database information 48
network topology layout
algorithms 60
symbol characteristics 44
symbols recognized, internet submap 69
testing event correlation rulesets 139
threshold Agent Policy Manager
definition
changing 231
creating 228
creating from an existing definition 228
defining
dialog box fields, description 237
procedure 235
deleting 232
diagnosing problems 264
distributing
distribution failures 230
procedure 229
successful distribution 230
how to define and distribute 221
modifying 231
viewing
active 229
pending 232
thresholds ruleset node 132
thresholds, setting 146
time, changing for status check 167
Tivoli Enterprise Console
forwarding events to 110
event adapter, controlling 113
event format 112
Tivoli NetView
changing
colors 100
fonts 100
sizes 100
tool palette 98
topology discovery daemons 7
tralertd daemon 16
trap settings ruleset node 134
trap-to-alert conversion 151
trapd daemon 13
trappgend daemon 14
traps 281
  creating correlation rulesets 136
  customizing
    conditions 154
    scenario 154
    verifying 156
  filtering 142
  typeset conventions  x
  Types of SmartSets  84

U
understanding MIBs
  browsing 195
  loading and unloading 193
unloading MIBs 195
unmanaged nodes, suppressing events from 120
unmanaging manager-container associations 190
updating host name
  local maps  50
  NFS-mounted maps  50
user plane 61
user profiles management dialog 30
user profiles, security
  creating 29
  deleting 31
  valid characters for names and passwords 31
  viewing 31
using layout
  automatic 97
  redo 96
using predefined MIB applications 198
using seed files
  backup manager 185
  network discovery 165
Utilities
  nvdbformat 19
  nvdbimport 18

V
verifying security permission for shell scripts  42
vfy_access command 42
viewing
  Agent Policy Manager definitions
    active 229
    pending 232
  event log 121
  group profiles, security 34
  routing table information 175
  security registration files 28
  user profiles, security 31

W
windows
  graphical interface, changing the size, fonts, and colors 100
  main, starting as icon 100
  warning for events 158
  workspace
    changing default 100
    creating automatically 119
    dynamic 119
    filtering events 148
    writing reports 217

X
X11 monochrome bitmap format
  (XBM) 95
XBM 95
  xnmappmon application 6
  xnmbrowser application 4
  xnmbuilder application 5
  xnmcollect application 5
  xnmfault application 5
  xnmgraph application 5
  xnmloadmib application 4
  xnmrunreport application 5
  xnmssnpconf application 4
  xnmtrap application 4
  xmap application
    defining 67
    description 3
    presenting information 67
    protocol switching 67

Z
z/OS systems
  discovering 165
  OVwDB fields 165
  zoom, quick 60
  zooming submaps 59