Contents

Preface ............................................................... ix
Who Should Read This Guide ................................. ix
Prerequisite and Related Documents ....................... ix
What This Guide Contains .................................... ix
Typeface Conventions .......................................... x
Online Information ............................................. x
Accessibility Information ..................................... xi
  Keyboard Access ............................................. xi
Contacting Tivoli Support ................................... xi

Chapter 1. Installing the NetView Program .................. 1
Installing the NetView Program ............................... 1
Installation Problems .......................................... 1
  Insufficient Disk Space .................................... 1
  TCP/IP Settings ........................................... 1
  Installing from Network Neighborhood .................. 1
  Client/Server Installations ............................... 1
  Additional Sources of Information ....................... 3
Changing the NetView Settings ................................ 3
Tivoli Enterprise Console® Adapter Configuration Application .... 3
Manually Configuring Tivoli Enterprise Console Integration .... 3
  SmartSet Filtering ....................................... 6
  Tracing and Logging .................................... 6
Web-Based Enterprise Management (WBEM) Support ........... 6
  Discovery ................................................. 6
  Object Properties ..................................... 7
  Dynamic Queries ....................................... 7
  Extensible .............................................. 7
Removing the NetView Program ................................ 7

Chapter 2. Getting Started .................................... 9
Starting the NetView Program ............................... 9
If the NetView Program Does Not Start ................... 10

Chapter 3. Getting Acquainted with the NetView Program .... 13
Using the Mouse .................................................. 13
Understanding the Basics of the NetView Console .......... 14
Submap Window .................................................. 14
  NetView Console ....................................... 15
  Menu Bar .............................................. 16
  Submap Stack ......................................... 17
  New Objects Area ..................................... 18
  Child Submap Area .................................... 18
  Status Bar ............................................ 19
Submap Explorer ................................................ 19
NetView Toolbars ............................................... 20
  NetView Standard Toolbar ............................ 20
  NetView Application Toolbar .......................... 22
  Status Filter Toolbar .................................. 23
Using the Navigation Tree Window ......................... 23
Saving the NetView Windows Layout ....................... 24
Customizing the NetView Console .......................... 24
Accessing Online Information ............................... 25

iii
Preface

This document provides a foundation for understanding and operating the Tivoli® NetView for Windows program. This product is referred to as the NetView program in this document.

This document explains the functions of the NetView program and how to use NetView to manage and monitor an open systems network.

Who Should Read This Guide

This document is intended for network operators and administrators.

Prerequisite and Related Documents

The following is a list of Tivoli NetView for Windows related publications:
- Tivoli NetView for Windows Programmer’s Guide
- Tivoli NetView for Windows Programmer’s Reference

What This Guide Contains

This book contains the following information:

- "Chapter 1. Installing the NetView Program" on page 1
  Explains how to install the NetView program.
- "Chapter 2. Getting Started" on page 9
  Explains how to start and stop the NetView program and the NetView daemons.
- "Chapter 3. Getting Acquainted with the NetView Program" on page 13
  Explains how to navigate in the NetView windows and use the online help.
- "Chapter 4. Understanding Client/Server" on page 29
  Explains how NetView can be installed and used on client machines.
- "Chapter 5. Working with the NetView Program" on page 31
  Describes the main features of the NetView program and tells how to use them.
- "Chapter 6. Working with Events and Event Filters" on page 61
  Explains how to view and interpret network events and how to create event filters.
- "Chapter 7. Managing the Network" on page 73
  Explains how to manage the MIB database and create MIB applications that can run with the NetView program.
- "Chapter 8. Managing Network Configuration" on page 85
  Describes network configuration and network polling tasks.
  Explains how to use the NetView program to diagnose and solve problems that can occur in your network.
- "Chapter 10. Diagnosing and Solving NetView Problems" on page 113
  Explains how to diagnose and solve problems that can occur with the NetView program.
- "Chapter 11. Maintenance" on page 139
  Describes tasks NetView administrators will need to perform.
• "Appendix A. System Performance Considerations for Large Networks" on page 141
  Explains maintenance items and performance considerations for large networks.

• "Appendix B. Installing NetView Using Microsoft System Management Server (SMS)" on page 145
  Describes how to install NetView using the Microsoft® System Management Server (SMS).

The glossary at the end of this document can assist you with terminology. To view additional terminology lists, refer to:

---

**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.

**ALL CAPS**
- Tivoli NetView for OS/390® commands appear in **ALL CAPS**.

---

**Online Information**

Click on the **Read Me** icon in the NetView Program Group for more information about the NetView program.

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 accessisible from the NetView Console using the **Help..Books Online** menu item, which will bring up 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 NetView, replace the `c` subdirectory with the appropriate locale specifier.

In addition, you can access online documents at this Web address:
  http://www.tivoli.com/support/

A user name and password are required.
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:

- Use assistive technologies such as screen-reader software and a digital speech synthesizer to hear what is displayed on the screen.
- Operate specific or equivalent features using only the keyboard.
- Magnify what is displayed on your screen using screen magnifiers.
- Customize display attributes such as color, contrast, and font size.
- Keyboard Navigation of the User Interface.

Assistive technology products such as screen-readers, function with both the text-based and graphical user interfaces found in this product. Consult the assistive technology product documentation for specific information about using it to access text-based (terminal emulation) or graphical interfaces.

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.

Contacting Tivoli Support

If you have a problem with any Tivoli product, you can contact Tivoli Customer Support. See the Tivoli Customer Support Handbook at the following Web address:

http://www.tivoli.com/support/handbook/

The handbook provides information about how to contact Tivoli Customer 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 you are in
- What information you should gather before contacting support
Chapter 1. Installing the NetView Program

Installing the NetView Program

Refer to the Release Notes for installation information.

Installation Problems

This section describes potential installation problems, what to do if these problems occur, and where to look for more information.

Insufficient Disk Space

If you have problems installing this program, first verify that there is sufficient disk space and paging space. To check the paging space, select the System icon from the Control Panel and then click the Performance tab to check the virtual memory.

Refer to the Release Notes for disk space and paging space information.

TCP/IP Settings

Check that your TCP/IP settings are correctly configured for IP Address, Subnet Mask, and Domain Name.

Installing from Network Neighborhood

When you install from another machine instead of from the CD media, it is important to mount the remote share to a drive letter.

Client/Server Installations

This section describes problems you may encounter when installing a NetView Server and clients, and how to resolve these problems. It also provides general information about the client/server installation.

NetView Password

The NetView program requires a NetView account on the server and on the client. NetView accounts are created during installation. During both the server installation and the client installation, the installation process prompts for a password. The password must be the same on both the server and the client to enable communication between the client and the server.

NetView Share

When selecting a drive on which to install the NetView Server, consider that the server exports the entire drive (\), rather than only \usr\ov\ . The client mounts the NetView share using the drive letter in the environment variable NV_MOUNT_DRIVE. When logging onto the client using an account other than the one used to install NetView, the share must be mounted using this drive letter. In addition, the NetView account and password must be the same on both the server and client.

If you are running the NetView console on a NetView client and the file share for the NetView Server is not mounted, you will receive the following message:

The server file share \myserver\NetView is not mounted. Mount it on drive X: using the NetView account password.
To correct this situation, do the following:

1. Run Windows Explorer.
2. Select Tools→Map Network Drive....

Complete the dialog box as follows:

1. For the Drive, choose the drive letter specified in the message.
2. For the Path, enter the file share specified in the message (myserver is used here as an example).

\myserver\NetView
3. Click OK.
4. When prompted for the password, enter the password for the NetView account on the server machine.

Environment Variables
The environment variables used by the client and defined by the installation are:
NV_MOUNT_DRIVE, TRAPD_CONF, NVSNMP_CONF, SNMPCOLLECTDIR, SNMPCOL_CONF, MIBDB, MIBFILES.

IP Name Resolution
Both the NetView client and the NetView Server need to be able to resolve the IP address and name of the other, including deriving the NetBIOS name (computer name). Currently, NetView derives the NetBIOS name from the resolved IP address. If the address, for example, 16.1.0.23 resolves using DNS to netview.tivoli.com, the NetBIOS name is assumed to be NETVIEW.

To continue this example, if the NetBIOS name is MYNODE instead of NETVIEW, the following message would be displayed from the client installation:
Unable to connect to NetView Server sharename \NETVIEW\NetView

To avoid this problem, if the NetBIOS name is different, add an alias in the HOSTS file, (%SystemDrive%\system32\drivers\etc\HOSTS) similar to the following:
16.1.0.23 MYNODE

Add aliases on the server for each client whose computer name is not the same as the one derived from the IP name.

IP address and name resolution is important for communication between the NetView client and the server and can cause problems if unexpected resolutions occur.

To aid in diagnosing such problems, use the \usr\ova\bin\host.exe utility to see how NetView is resolving IP addresses and names. If successful, host.exe displays the name resolution from the address and the reverse resolution from the name. Determine whether the IP addresses and names are what you expect. Using the /s switch, the host.exe utility can use net send to send a pop-up box to the machine to verify that the NetBIOS name is for the correct machine.

host.exe [/s] [IP address | IP name | NetBIOS name]

With no parameters, the host.exe utility gets the local address and resolves it. You can use the host.exe utility from the server to verify IP addresses for all the clients.

NetView Cannot Connect to Share
Sometimes, the NetView client cannot connect to the NetView share on the server. This may be due to an existing connection on the client machine to a drive on the server, using another account on the server. If this is the case, disconnect that...
share on the client and reinstall NetView. On Windows, a machine cannot access two different shares using two different accounts on the same remote machine.

In addition, the server and the client must be using either the same locale or must be using compatible locales for a NetView client to connect to the NetView Server. If the locale on the client does not match the locale on the server, the client cannot connect unless the server is running the U.S. English locale (enu), which is the default locale. Version checking is also performed to prevent older NetView clients and servers from interacting; this extra checking is required for multiple language support.

**Additional Sources of Information**

If you have questions concerning the NetView program, which are not described in the previous sections, check the following sources for more information.

- NetView installation log (available from the NetView Installation group icon).
- "If the NetView Program Does Not Start" on page 10
- "Diagnosing Problems with the NetView Program" on page 134

If you are unable to resolve the problem, call your Tivoli customer support.

**Changing the NetView Settings**

The NetView program enables you to change the NetView windows and some daemon information. For example, you may want to discover a subset of gateway nodes instead of all the nodes in your network. To change the NetView settings, click on the main menu. Refer to the online help for additional information.

**Tivoli Enterprise Console® Adapter Configuration Application**

This application enables configuration of the Tivoli Enterprise Console Adapter for NetView (which provides integration between the NetView program and the Tivoli Event Console). To start this application, run \usr\ov\bin\TecConfig.bat.

After running the Configuration Application, complete step 6 on page 4 and step 8 on page 5 in to finish configuring integration with the Tivoli Enterprise Console.

**Manually Configuring Tivoli Enterprise Console Integration**

This section describes the steps to manually install and configure the Tivoli Enterprise Console Event Adapter for the Tivoli NetView program. If you used the application described in Application, complete step 6 on page 4 and step 8 on page 5 to finish configuring the Tivoli Enterprise Console Integration.

1. Decide which version of the adapter you will use: the TME® or non-TME version. If your Windows workstation has the Tivoli Framework installed, then follow the steps labeled for TME. If the Tivoli Framework is not installed, follow the steps labeled for non-TME. If you are not sure whether the Tivoli Framework is installed, check for the subdirectory \Tivoli\bin\win32–ix86\TME. If this subdirectory exists, the Tivoli Framework is installed.
2. Copy the appropriate executable from the \usr\ov\bin directory:
For... Enter...

TME

```bash
copy tecad_nv6k_tme.exe tecad_nv6k.exe
```

non-TME

```bash
copy tecad_nv6k_non_tme.exe tecad_nv6k.exe
```

3. Register with NetView by entering the following command in a command window:

```bash
ovaddobj \usr\ov\lrf\tecad_nv6k.lrf
```

4. Identify the event server to be used by editing the `\usr\ov\conf\tecad_nv6k.conf` file.

For... The ServerLocation entry format is...

TME

```bash
ServerLocation=Event Server
```

non-TME

ServerLocation=<hostname>, where <hostname> is the name of the host running the Tivoli Enterprise Console. Also, specify the ServerPort entry for non-TME in the following format:

```bash
ServerPort=<number>
```

Where <number> is the port number used by the Tivoli Enterprise Console event server to listen for events. Specify 0 (zero) if the event server is using portmapper. (This is typically the case.)

5. The Tivoli Enterprise Console Adapter uses the following files on a NetView Server:

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tecad_nv6k.exe</td>
<td>Is the executable.</td>
</tr>
<tr>
<td>tecad_nv6k.conf</td>
<td>Is the global configuration file.</td>
</tr>
<tr>
<td>tecad_nv6k.err</td>
<td>Contains the tracing options.</td>
</tr>
<tr>
<td>tecad_nv6k.cds</td>
<td>Contains event class mapping specifications.</td>
</tr>
<tr>
<td>tecad_nv6k.lrf</td>
<td>Is the NetView registration file.</td>
</tr>
<tr>
<td>tecad_nv6k.oid</td>
<td>Contains the mapping of SNMP object identifiers to names.</td>
</tr>
<tr>
<td>tecSmartSetFilter.Conf</td>
<td>Is the SmartSet filtering configuration file.</td>
</tr>
</tbody>
</table>

You can modify these files to add or remove events sent to the event server.

6. Customize the Tivoli Enterprise Console to understand NetView events. Perform these steps on your Tivoli Enterprise Console server. Refer to the documentation for more information.

a. Copy the following files in the `\usr\ov\conf` directory to the system running the Tivoli Enterprise Console:
   - `tecad_snmp.baroc`
   - `tecad_nv6k.baroc`
   - `tecad_ov.baroc`
   - `ov_default.rls`
b. If you do not have a default rule base for your environment, create one with the following command:
   `wcrtrb -d <directory> My_rb`

c. Copy the Tivoli Enterprise Console-supplied default rule base (named Default) into your rule base. Enter the following command:
   `wcprb -cr -f Default My_rb`

d. Import the event classes into the rule base you will be using. Enter the following sequence of commands:
   `wimprbclass tecad_snmp.baroc My_rb`
   `wimprbclass tecad_ov.baroc My_rb`
   `wimprbclass tecad_nv6k.baroc My_rb`

e. Import the rules into the rule base. Enter the following command:
   `wimprbrules ov_default.rls My_rb`

f. Compile the rule base. Enter the following command:
   `wcomprules My_rb`

g. Restart the event server to load the new rule base. Enter the following sequence of commands:
   `wstopesvr`
   `wloadrb My_rb`
   `wstartesvr`

7. If you want to filter traps by SmartSet, edit the file
   `\usr\ov\conf\tecSmartsetFilter.conf`; (refer to the comments for syntax information.) Traps are first checked against the normal filter and then checked for membership in each SmartSet listed for that trap in the tecSmartsetFilter.conf file. The node generating the trap must be a member of one of the listed SmartSets for the trap to be forwarded to the Tivoli Event Console. If the trap is not listed in the tecSmartsetFilter.conf file, it is not checked for SmartSet membership. SmartSet filtering can be done on traps of any enterprise, including generic traps. The adapter is sensitive to dynamic changes in SmartSet membership.

   For more descriptive information on SmartSet filtering, see "SmartSet Filtering" on page 6. For more information about Tracing and Logging see "Tracing and Logging" on page 6.

8. For TME users only, complete these steps:
   a. Add the relative `$BINDIR\bin` path to the system environment, run:
      `&windir\system32\drivers\etc\Tivoli\setup_env`

      Then enter the following commands:
      `bash`
      `echo $BINDIR`

      Set the path in the System dialog from the Control Panel.

   b. Restart the machine.

   c. On the TMR server, create an administrator with the Label, User Login Name, and Group Name of NetView (assuming NetView is the name of the account the user logs in under). Edit logins to include:
      `NetView@hostname_of_machine_running_Netview_Server`

      Set the Resource Roles for Event Server to have User privileges.

9. Start the event adapter and test. If Tivoli NetView has been restarted, the event adapter is started automatically. Otherwise, start the event adapter by entering the following command:
ovstart tecad_nv6k

Send a sample node down event for a host of your choice:

```
event -e NDWN_EV -h <hostname>
```

A node down event is displayed in the Tivoli Enterprise Console Events Display. If an event does not display, check the following:

- That the event server is identified properly in the 
  `\usr\ov\conf\tecad_nv6k.conf` file
- That the Event Server processed the event classes and rules successfully

## SmartSet Filtering

SmartSet filtering acts as a prefilter to the Tivoli Enterprise Console Adapter on Windows and is based on nodes. You should only use SmartSets that have nodes as members. For the purposes of SmartSet filtering, the node name is extracted from the second varbind for NetView enterprise traps, and from the agent address field for all others. Keep this in mind when selecting NetView traps.

All traps pass the SmartSet prefilter, by default. A trap will only be filtered out at this stage if:

- there is an entry in `tecSmartsetFilter.conf` for it
- the entry has one or more SmartSets listed against it
- the node associated with the trap is not a member of any of the listed SmartSets

See the file `\usr\ov\conf\tecSmartsetFilter.conf` for details of the syntax.

Once the trap has passed the filter it is then subject to the rules specified in the standard Tivoli Enterprise Console Adapter configuration files.

## Tracing and Logging

To diagnose a problem, edit `\usr\ov\conf\tecad_nv6k.err` to enable logging for the DRVSPEC module by specifying a file name against each level. Note that serious issues starting the daemon will always be logged in the NetView log file, `\usr\ov\log\nvtecad.log`. The tecad_nv6k.err file by default now lists this file for FATAL, MAJOR, and MINOR level errors.

Turn on the NORMAL level to see the results of parsing the tecSmartsetFilter.conf file, and to see the membership of each SmartSet at startup and when it changes.

## Web-Based Enterprise Management (WBEM) Support

The NetView program provides integrated support for Microsoft’s WBEM initiative. WBEM is included in Windows 2000. For Windows, download and install the WBEM kit from [http://wbem.freerange.com](http://wbem.freerange.com). For full details of NetView WBEM default support and how it can be extended, refer to the document in the `\usr\ov\doc\wbemreadme.htm` file.

NetView WBEM support includes the following:

### Discovery

The NetView program can discover nodes running WBEM and create a Service SmartSet for these WBEM providers.
Object Properties

By default, the NetView program queries a set of system and network data and stores the results in the object database which can be viewed from the Other tab page in Object Properties. This data can be updated regularly using nvsniffer and manually, using Tools -> WBEM -> Demand Poll.

Dynamic Queries

You can use the NetView program to dynamically query WBEM providers for predefined data. Use the default queries under Tools -> WBEM or create your own query.

Extensible

All the above mechanisms may be extended for user-defined queries.

Removing the NetView Program

Use one of the following methods to uninstall the NetView program:

- From the start menu, select Programs -> Tivoli NetView -> Installation -> Uninstall NetView
- From the Control Panel, click Add/Remove Programs, select Tivoli NetView from the list of installed software, and press Add/Remove.

The NetView program is removed from your system

Notes:

1. At the end of the removal process, if you click on the Details button, you may be told that some directories could not be removed. However, the uninstaller makes every attempt to clear the NetView program from the system, and you may find that these directories are removed.

2. After uninstalling the NetView program, all NetView environment variables are removed, however, these environment variables continue to be defined until the system is restarted.

3. If you will be reinstalling the NetView program, you must restart the system.
Welcome to the NetView program and the world of effective network management. The NetView program puts you in control of your network by providing configuration, fault, and performance management functions for your network resources.

This chapter provides information about:

- "Starting the NetView Program"
- "If the NetView Program Does Not Start" on page 10

## Starting the NetView Program

To start the NetView program, select **NetView Console** from the NetView Program group. When the NetView program starts, it displays the NetView console, a Tip of the Day, and the Submap Explorer.

**Figure 1** displays the NetView Console.

![NetView Console Displayed at Startup](image)

**Figure 1. NetView Console Displayed at Startup**

**Figure 2 on page 10** displays the Tip of the Day.

![Tip of the Day](image)
If the NetView Program Does Not Start

If the NetView program does not start after you double-click on the NetView icon:

- Check the NetView message log (/usr/ov/log/nv.log) for error messages.
- Ensure all the daemons are running properly. See "Checking Daemon Status" on page 114.
• See "Diagnosing Problems with the NetView Program" on page 134 for information on diagnosing problems with the NetView program.
• Check the NT application event log for errors whose source is NV.
Chapter 3. Getting Acquainted with the NetView Program

The NetView program displays a logical representation of your network through dynamic topology maps that contain hierarchies of submaps. The submaps include graphical representations of your network at several levels. These levels include root, internet, network, segment, and node. To access submaps that are lower in the map hierarchy, you double-click on symbols that explode into lower submaps.

The submaps support several network layout algorithms, including row/column, point-to-point, bus, star, ring, tree, and none. If you decide not to select a layout algorithm (none), you can place the objects in any layout you prefer.

The NetView program conforms to Microsoft Windows standards. If you are not familiar with Windows conventions, read this chapter. This chapter briefly describes the use of the mouse and window manipulation.

- "Using the Mouse" on page 13
- "Understanding the Basics of the NetView Console" on page 14
- "NetView Toolbars" on page 20
- "Using the Navigation Tree Window" on page 23
- "Saving the NetView Windows Layout" on page 24
- "Customizing the NetView Console" on page 24
- "Accessing Online Information" on page 25
- "Using Menus" on page 25
- "Changing Tables in Dialog Boxes" on page 27

If you are familiar with Windows conventions, you might want to go directly to "Understanding the Basics of the NetView Console" on page 14.

Using the Mouse

You can use any 2-button mouse to work with the NetView program. By default, the button on the left side is button 1, and the mouse button on the right side is button 2. If you have a 3-button mouse, the middle button is not used. If the mouse is configured for a left-handed person, the buttons are reversed.

There are three general functions that you can perform with a mouse: selecting, dragging, and performing custom operations. Table 1 identifies which mouse buttons to click to perform each of these operations.

Table 1. Using a Two-Button Mouse

<table>
<thead>
<tr>
<th>Mouse Action</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clicking button 1</td>
<td>Selects one object on a submap or item on a menu. With this action, you can select only one object at a time.</td>
</tr>
<tr>
<td>Double-clicking button 1</td>
<td>Performs the default action. Depending on where you click, this could open a child submap, execute a symbol’s operation, or create a new submap.</td>
</tr>
<tr>
<td>Holding the Ctrl key and clicking button 1</td>
<td>Selects multiple objects. Use mouse button 1 to select the first item, then hold down the Ctrl key and click mouse button 1 on each additional item to add it to the selection list.</td>
</tr>
<tr>
<td>Holding and dragging button 1 on an object</td>
<td>Drags and drops an object. Use this action to move an object to a new position on the submap.</td>
</tr>
<tr>
<td>Clicking button 2</td>
<td>Activates the pop-up context menu for the symbol.</td>
</tr>
</tbody>
</table>
Table 1. Using a Two-Button Mouse (continued)

<table>
<thead>
<tr>
<th>Mouse Action</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Holding and dragging button 1 on background</td>
<td>Selects a rectangular area. Press and hold mouse button 1 on a blank area of the window and then drag mouse button 1 over the area of the window containing icons you want to work with. This enables you to select a group of icons.</td>
</tr>
<tr>
<td>Holding Ctrl key and dragging mouse button 1 on background</td>
<td>Selects multiple rectangular areas. Do this to keep any areas previously selected and add a new area to the selection list.</td>
</tr>
<tr>
<td>Holding Shift key and clicking mouse button 1</td>
<td>Works the same as holding the Ctrl key and dragging mouse button 1 on the background.</td>
</tr>
<tr>
<td>Holding Shift key and double-clicking mouse button 1</td>
<td>Opens a child submap in a new window.</td>
</tr>
</tbody>
</table>

Understanding the Basics of the NetView Console

In a Windows environment, your screen can contain multiple windows, but you can interact with only one window at a time. The window you interact with is known as the active window.

Windows are identified by title bars located at the top of the window border. Menu choices are listed in menu bars located under the title bar. To select a menu, choose mouse button 1.

The NetView console uses all the standard window-handling features of the Windows operating system. For example, you can change the size and location of a NetView window on your screen. For more information about working with windows, see Windows help.

Submap Window

The submap window contains a graphical representation of a portion of your network. Using this window, you can modify submaps by adding, moving, changing, and deleting objects and symbols from various submaps. Use these operations to modify topology maps to represent information that cannot be dynamically discovered. See Figure 4 on page 13.
NetView Console

Several features have been added to make the console more useful. These include:

- **Submap Sorting**
  Select **Submap -> Sort by..** to see a list of object properties that can be used to sort the symbols. For example, Label, IP Address, Status and many others. This can be useful to quickly find a particular node or see holes in the IP address ranges.

- **Property Tips**
  A property tip is a snippet of information that is displayed when you pass the cursor over a symbol in the submap. NetView provides a number of property tips enabling you to quickly access various types of information about an object. Use the drop down list in the NetView Console to select a different property tip. Obtaining some property tips can be slow, depending on the underlying mechanism used to gather the information.

  The following types of property tips are supported:
  - NetView map database fields (Label, Status)
  - NetView object database fields (Selection Name)
  - MIB Object IDs or MIB Expressions (System Contact), obtained using SNMP
  - NetView ODBC database fields (Up Time)
  - TCL scripts (WBEM, Health)

  You can add new property tips by editing the `\usr\ov\conf\c\tips.conf` file (if you have installed a non-English version of NetView, replace the `c` subdirectory with the appropriate locale specifier). Each entry in this file consists of three fields:
  - The label of the Property Tip followed by an equal sign (=)
  - The type of property: MAP, OVWDB, MIB, ODBC, or TCL
  - The property to be displayed

  For information about the syntax of this file, refer to the comments at the beginning of the file.
• Hide Symbols Matching a Status Color

   Click the palette of status colors on the Tool Bar to hide all the symbols of that status color. Click again to redisplay them.

• Submap Bookmarks

   The View → Bookmark option enables you to place a marker on a submap. To place a bookmark on a submap:
   1. Go to the submap you want to mark.
   2. Select View → Bookmark.
   3. In the Submap Bookmarks dialog, click on Add to create a bookmark.
   4. Click Close.

In the Submap Bookmarks dialog, you can also remove a marker and go to the submap. To go to a submap, double-click on the submap name or click on the name and then select Open.

• IntelliMouse Horizontal Scrolling

   In addition to the IntelliMouse features documented in the NetView help, you can scroll a zoomed submap horizontally. To do this, hold down the Ctrl and Shift keys while simultaneously scrolling the mousewheel. If you move the mousewheel up, the submap is moved to the left. If you move the mousewheel down, the submap is moved to the right. For more information on IntelliMouse, select Help -- Help Topics -- Index tab. In the Index tab, search for IntelliMouse.

**Menu Bar**

The menu bar contains pull-down menus for accessing and controlling network management operations. The NetView program provides Beginner and Advanced menus. The Beginner menus are a subset of the NetView menus and help the user responsible for monitoring and managing networks. The Advanced menus are the complete set of NetView menus and help the user who is responsible for configuring NetView and writing network tools. When the NetView program first starts, it displays the Beginner menu. If you want to switch to the Advanced menus, select Options..Console Settings. The following table provides an overview of NetView menus.

<table>
<thead>
<tr>
<th>Menu Bar</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>Create, open, and delete maps and snapshots, and exit the NetView program.</td>
</tr>
<tr>
<td>Edit</td>
<td>Modify the properties of maps, submaps, and snapshots; highlight and select objects; and find objects.</td>
</tr>
<tr>
<td>Object</td>
<td>Create new objects and connections; open, delete, manage or not manage, acknowledge or not acknowledge existing objects, discover objects, and modify object properties.</td>
</tr>
<tr>
<td>Submap</td>
<td>Create new submaps; delete or refresh submaps; enable or disable automatic layout; and shadow user objects.</td>
</tr>
<tr>
<td>View</td>
<td>Display different aspects of a map and the network by opening and viewing different submaps; display critical objects.</td>
</tr>
</tbody>
</table>
Monitor
Monitor pertinent information about the system, network, and devices on the network, and graph network information.

Test
Perform tests on selected objects in the network.

Tools
Manage information such as Management Information Base (MIB) values, MIB collections, and MIB applications.

Options
Customize the NetView program, including defining SNMP traps, configuring SNMP nodes, defining filters for events, and setting user preferences.

Window
Display and organize NetView windows.

Help
Access help for the NetView program and any related applications.

You can use the keys on the keyboard to accomplish a specified task instead of clicking on the menu bar or pull-down menu items.

To activate the menu bar, press **F10** or press **Alt**. Under certain circumstances, some menu items appear grayed. This is a visual indication that the choice is disabled and cannot be accessed. For example, if you are viewing a read-only map or snapshot, you are not given the choice of deleting objects.

**Submap Stack**

The submap stack is located on the left side of the submap window. This area displays a stack of icons representing the parent submaps that you have already displayed. It shows the hierarchy of submaps you have opened for a particular map. Each time you open an object, a submap icon is added to the stack. The bottom icon in the stack shows you the parent of an object represented in the current submap. See [Figure 5].

![Figure 5. Submap Stack](image)
The bottom left corner of the submap stack displays the icon for the submap that is currently open in the submap window. See Figure 6 for an example.

![Figure 6. Submap Icon](image)

New Objects Area

The New Objects area is displayed only when automatic layout is disabled for the submap. The New Objects area is located at the bottom of the submap window and holds symbols for newly discovered objects. See Figure 7.

These symbols remain in the New Objects area until you drag and drop them onto a submap, until automatic layout is enabled, or until you have selected Submap..Refresh Layout. When a map is refreshed with automatic layout off, the newly discovered objects are added to the appropriate submap.

![Figure 7. New Objects Area](image)

Child Submap Area

The child submap area is located at the bottom of the submap window, below the New Objects area. The child submap area shows the submaps that you have previously opened from the current submap. You can open a submap from this area, or bring it into view if it is already opened in another window on the screen. See Figure 8.

![Figure 8. Child Submap Area](image)
Status Bar

The status bar is located at the bottom of the submap window.

![Status Bar](image)

*Figure 9. Status Bar*

The following information may appear:

- Descriptions of menu items and toolbar icons.
- Messages that indicate the status of the NetView program or of integrated applications.
- The count of the number of objects in the database.
- The Network Monitor Activity Bars.
- The zoom factor, if one is being used.
- The current time.

Submap Explorer

The Submap Explorer is displayed by default when the console is displayed. If it is closed, it can be brought back by clicking on Window...Submap Explorer. The title of the window is Exploring followed by the name of the submap displayed. The Submap Explorer displays one submap at a time just like a regular submap window. As the Windows Explorer is used to explore files, the Submap Explorer is used to explore submaps. The left side is a tree view of the submaps that is used to navigate the submap hierarchy. Each submap is represented by its parent object. Double-clicking an object or clicking the + symbol next to an object will expand it to display the next level. Then, an object at the next level can be expanded and so on. In this way, the entire hierarchy can be navigated. Double-clicking or clicking the – symbol will collapse an expanded object. Selecting an object will display the contents of its submap in the right-hand side of the Submap Explorer. The shortcut menus are available for objects in the list or tree view by right clicking the object.

The right side is a list view displaying the contents of the submap selected in the tree view. It displays all of the objects in the submap and their properties. The set of properties displayed is chosen from the dropdown list in the toolbar. The dropdown list displays Property Sets read from \usr\ov\conf\c\explore.conf at the startup of the console. If you have installed a non–English version of NetView, replace c with the appropriate locale specifier.

Double-clicking an object opens up its child submap and displays it in the list view. Resting the cursor over any of the icons in the toolbar will display the tooltip for that icon. The icons are as follows:

<table>
<thead>
<tr>
<th>Icons</th>
<th>Tooltip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save</td>
<td>Save the current view in HTML or text format.</td>
</tr>
<tr>
<td>Copy</td>
<td>Copy to the clipboard.</td>
</tr>
<tr>
<td>Large Icons</td>
<td>Switches to large icons, which shows more detail in the icons.</td>
</tr>
</tbody>
</table>

Table 2. Icon Tooltips
Table 2. Icon Tooltips (continued)

<table>
<thead>
<tr>
<th>Icons</th>
<th>Tooltip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small Icons</td>
<td>Switches to small icons, which enables more objects to be displayed at one time without scrolling.</td>
</tr>
<tr>
<td>Filter</td>
<td>Filter Availability data based on time periods. To use this icon, the Availability View must be selected.</td>
</tr>
<tr>
<td>Refresh</td>
<td>Refresh the current view.</td>
</tr>
</tbody>
</table>

The width of each column can be changed by dragging the column separators. The list can be sorted by the contents of a column by clicking on the column header. The first click sorts the list in ascending order, the second in descending order.

**NetView Toolbars**

The NetView toolbars provide convenient access to applications and tools you regularly use. For example, you can open a new application, check SNMP and interface traffic, and print the contents of a particular window.

Toolbars provide shortcuts to menu items for commonly used commands. The NetView program provides Beginner and Advanced toolbars to accompany the Beginner and Advanced menus. The Beginner toolbar, like the Beginner menu, is a subset of the complete NetView toolbar and helps the user responsible for monitoring and managing networks. The Advanced toolbar is the complete NetView toolbar and helps the user responsible for configuring NetView and writing network tools. When the NetView program first starts, it displays Beginner toolbars. If you want to switch to the Advanced toolbars, select Options..Console Settings.

The NetView program displays three toolbars: the NetView Standard toolbar, the NetView Application toolbar, and the Status Filter toolbar.

You can move these toolbars anywhere on the desktop. You can also choose not to display them from the main menu by selecting the Window command, then the toolbar.

Each icon in the NetView toolbars represents a menu item, an available tool, an application, or a status. These tools and applications allow you to perform tasks on selected objects in a submap or on all the information in a submap. You can click on an icon to perform an action or to start that tool or application. If an icon is grayed, it cannot be selected.

**NetView Standard Toolbar**

The applications and tools provided by the NetView Standard toolbar are described in the NetView Standard Toolbar.

Table 3. NetView Standard Toolbar

<table>
<thead>
<tr>
<th>Icon</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>![New Map Icon]</td>
<td>The New Map icon enables you to create a new map.</td>
</tr>
</tbody>
</table>
Table 3. NetView Standard Toolbar (continued)

<table>
<thead>
<tr>
<th>Icon</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Open Map Icon" /></td>
<td>The Open Map icon enables you to open an existing map and display its default submap.</td>
</tr>
<tr>
<td><img src="image" alt="Save As Icon" /></td>
<td>The Save As icon saves the current state of the open map as either a new map or a snapshot.</td>
</tr>
<tr>
<td><img src="image" alt="Find Name Icon" /></td>
<td>In the Find Name combo box, enter the object name or the label, and then press the Enter key or the Find Next icon. Select Options..Console Settings to configure whether the Find Name operation searches on the object name or on the label.</td>
</tr>
<tr>
<td><img src="image" alt="Find Next Icon" /></td>
<td>The Find Next icon finds the object that you previously entered in the Find Name combo box.</td>
</tr>
<tr>
<td><img src="image" alt="Find Object Icon" /></td>
<td>The Find Object icon displays the Find dialog box.</td>
</tr>
<tr>
<td><img src="image" alt="Browse MIBs Icon" /></td>
<td>The Browse MIBs icon enables you to browse the MIB properties for any object in the current submap.</td>
</tr>
<tr>
<td><img src="image" alt="Navigation Tree Icon" /></td>
<td>The Navigation Tree icon displays the Navigation Tree if it is not already displayed. If you select this icon when the Navigation Tree is already displayed, the Navigation Tree is restored and raised to the top.</td>
</tr>
<tr>
<td><img src="image" alt="Critical Objects Icon" /></td>
<td>The Critical Objects icon shows all the critical child objects for the object you select.</td>
</tr>
<tr>
<td><img src="image" alt="Object Properties Icon" /></td>
<td>The Object Properties icon brings up the Object..Object Properties dialog box for the selected object.</td>
</tr>
<tr>
<td><img src="image" alt="Zoom In Icon" /></td>
<td>The Zoom In icon enlarges the current submap.</td>
</tr>
</tbody>
</table>
Table 3. NetView Standard Toolbar (continued)

<table>
<thead>
<tr>
<th>Icon</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Zoom Out Icon" /></td>
<td>The Zoom Out icon shrinks the current submap.</td>
</tr>
<tr>
<td><img src="image" alt="Scale Map Icon" /></td>
<td>The Scale Map icon changes from a map that uses zooming to a scaled map, if the current submap view enables zooming. This feature scales the submap picture to fit your submap window. If the current view is already a scaled map when you select this icon, the view is set to zoom mode.</td>
</tr>
<tr>
<td><img src="image" alt="Property Tips Icon" /></td>
<td>Property Tips enable you to choose the information that is to be displayed when you move your cursor over an icon in a submap.</td>
</tr>
<tr>
<td><img src="image" alt="About NetView Icon" /></td>
<td>The About NetView icon displays the full name, version number, and copyright information for the NetView program.</td>
</tr>
</tbody>
</table>

### NetView Application Toolbar

The applications and tools provided by the NetView Application toolbar are described in Table 4.

Table 4. NetView Application Toolbar

<table>
<thead>
<tr>
<th>Icon</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Event List Icon" /></td>
<td>Displays the events generated in an SNMP-based network. The events are presented as lists or cards. This is a dynamic display of events as they occur in your network.</td>
</tr>
<tr>
<td><img src="image" alt="Ping Icon" /></td>
<td>Sends a ping to the node you specify, and returns the response.</td>
</tr>
<tr>
<td><img src="image" alt="Route Icon" /></td>
<td>Locates and displays the route between the two nodes you specify.</td>
</tr>
<tr>
<td><img src="image" alt="MIB Icon" /></td>
<td>Displays the MIB system values Description, Object ID, Up Time, Contact, and Name for the selected object.</td>
</tr>
<tr>
<td><img src="image" alt="Network Interface Icon" /></td>
<td>Displays the network interface values Index, Interface, Type, MTU, Status, InPackets, InErrors, OutPackets, and OutErrors for the selected object.</td>
</tr>
</tbody>
</table>
Table 4. NetView Application Toolbar (continued)

<table>
<thead>
<tr>
<th>Icon</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Graphs interface traffic data received from selected objects. You can use this graph of input and output error packets to gather performance information." /></td>
<td>Graphs interface traffic data received from selected objects. You can use this graph of input and output error packets to gather performance information.</td>
</tr>
<tr>
<td><img src="image" alt="Collects and displays MIB data for the node you specify." /></td>
<td>Collects and displays MIB data for the node you specify.</td>
</tr>
<tr>
<td><img src="image" alt="Displays a graph of data that was previously collected by the MIB data collector for the selected objects." /></td>
<td>Displays a graph of data that was previously collected by the MIB data collector for the selected objects.</td>
</tr>
</tbody>
</table>

**Status Filter Toolbar**

The Status Filter toolbar displays or hides the nodes in the submap. For example, click on the **Unknown Status** icon if you want to hide all unknown nodes. The Status Filter icons are displayed in **Figure 10**.

![Figure 10. Status Filter Toolbar Icons](image)

**Using the Navigation Tree Window**

In addition to Submap Stacks, the NetView program provides a Navigation Tree to help you move around submaps. To use the Navigation Tree, select **Windows..Tree**. The Navigation Tree window shows the hierarchy of the submaps that you have opened and helps you keep track of your current position in a particular submap hierarchy. Submaps are represented with icons placed in a tree-like structure with lines connecting the parent and child submaps.
When you start the NetView program, the first submap to appear in the tree is the root submap or the home submap that you have designated. Each time you open a new submap using the submap window, its corresponding icon is displayed in the Navigation Tree in the correct relationship to the other open submaps.

You can display a submap in the hierarchical structure of the open map by double-clicking on an icon in the Navigation Tree window. See Figure 11.

![Navigation Tree Window](image)

**Figure 11. Navigation Tree Window**

**Saving the NetView Windows Layout**

When you exit the NetView program, your current window settings are saved as the new defaults.

**Customizing the NetView Console**

The NetView program allows you to change many of the display settings, such as the fonts and colors used in all windows the program displays. You can also add your own menu items and specify how the Find operation will work.

To customize the display defaults select **Options..Console Settings** from the menu bar.
Accessing Online Information

A variety of information is available online including question and answer help, reference information, step-by-step procedures, and online user and programming guides.

To access the list of help topics, select Help..Help Topics from the main menu bar. For overall information on the dialog boxes, click on the Help button. For help on individual fields, click on the question mark (?) in the upper-right corner of the dialog box, then click on the field.

To access the NetView books, select Help..Books Online from the main menu.

Using Menus

NetView tasks can be started from menu bar items, shortcut menus, a NetView toolbar, or the Navigation Tree window. Each NetView application and user-added application selected from menu bar items runs in its own separate window. The NetView program contains the following types of menus:

- Pull-down
- Cascade
- Shortcut

These menus contain items that enable you to access and perform the operations of the NetView program. If a menu item is grayed, that choice is disabled because one of the following conditions exists:

- An object is not selected.
- The selected objects have capabilities that are incompatible with the menu item.
- The number of items selected is not within a range of minimum and maximum number of selections set for that menu item.
- The menu item concerns a map, but you are working with a snapshot.
- The menu item concerns a snapshot, but you are working with a map.
- The operation you requested is for a read-write map, but you are working with a read-only map.
- The operation you requested is exclusive to read-only maps, but you are working with a read-write map.

Accessing Pull-Down Menus and Cascade Menus

Pull-down menus are accessed by selecting an option from the menu bar. Cascade menus may appear when you select a menu item from a pull-down menu or a shortcut menu. For example, when you select All Submaps from the Object pull-down menu on the main menu bar, you see a cascade menu with the Cut, Copy, Hide and Unhide All Menu items.

Shortcut Menus

Shortcut menus are pop-up menus that are provided for performing operations on individual objects and submaps. The shortcut menus include operations from the menu bar.

To access shortcut menus, follow these steps:
1. Position the mouse pointer on the object or on the submap (background area).
2. Click mouse button 2.
A (pop-up) shortcut menu for the selected item is displayed.

3. Position the mouse pointer on a menu item and click mouse button 2 to select tasks from the shortcut menu.

   If you decide not to select an option from the shortcut menu, position the mouse pointer on an area outside of the shortcut menu and click mouse button 2 to close the shortcut menu.

Different shortcut menus appear as you click on different areas of the NetView windows. Table 5 describes the shortcut menu options.

*Table 5. Shortcut Menu Options*

<table>
<thead>
<tr>
<th>If you click on:</th>
<th>You see a shortcut pop-up menu with tasks for:</th>
<th>Shortcut menu options:</th>
</tr>
</thead>
<tbody>
<tr>
<td>A symbol or an object in a submap</td>
<td>Only that object or symbol</td>
<td>• Manage</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Unmanage</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Acknowledge</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Unacknowledge</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Open in New Window</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Cut</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Delete...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Explore...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Management Page...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Object Properties...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Show in...</td>
</tr>
<tr>
<td>The background of a submap</td>
<td>The entire submap</td>
<td>• Parent</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Home</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Root</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• SmartSets</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Paste</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Find...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Properties...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Browser...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Close</td>
</tr>
<tr>
<td>A submap from the Navigation Tree window</td>
<td>That submap and submap window</td>
<td>• Open</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Open in New Window</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Delete Branch</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Close Submap</td>
</tr>
<tr>
<td>The background of the Navigation Tree</td>
<td>The Navigation Tree</td>
<td>• Clean Up Tree</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Exit</td>
</tr>
<tr>
<td>The background of the Event Browser</td>
<td>The Event Browser</td>
<td>• Show Node on Map</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Node Properties</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Ping Node</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Run Command</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Event Details</td>
</tr>
</tbody>
</table>
Changing Tables in Dialog Boxes

The NetView program displays information in tables in several dialog boxes. You can change the order of the information in columns and you can change the size of a column.

- To sort the information from highest to lowest or ascending order, place the mouse cursor in the column head and click on mouse button 1.
- To sort the information from lowest to highest or descending order, place the mouse cursor in the column head and click mouse button 2.
- To change the size of a column, place the mouse cursor on the column head separators. When the mouse cursor changes, move the cursor to the left or right to widen or narrow the column.

Printing Tables in Dialog Boxes

You can print tables in dialog boxes by pressing the Ctrl–P key combination. For example, you can print the Locate Route table from a Test→Locate Route command or the IP Services table from a Monitor→Network→IP Services command.

Accessing NetView from a Web Browser

The NetView Web client provides a dynamic Web interface that allows you to view submaps, run diagnostic network tools, and view events as they come into the server.

To start the NetView Web client, follow these steps:

1. Point your Web browser to the following URL: http://myserver:8080/NetViewApplet.html (where myserver is the host name or IP address of the machine where the NetView server is running). A password window prompts you for your user ID and password.
2. Enter your user ID and password and click OK.

To view submaps, the NetView console must be running on the NetView server. To run the Web client in a browser, you may need a Java™ plugin. The Web client may also be run as a standalone Java application. Refer to the Release Notes for more information.
Chapter 4. Understanding Client/Server

The NetView program supports client/server in a distributed network environment. In this environment, the NetView console runs on separate client computers enabling you to distribute CPU and memory requirements to the client. Clients also maintain the map database. You can use several clients, enabling you to divide the management tasks among more operators at one time.

The server provides event and topology status information to clients and handles network discovery.

A NetView client works with map information stored on its own system and object and topology information.

NetView Client/Server Model

The NetView for NT Client/Server implementation implies a central map with some local map features.

Central Map Features

Basically, object changes are represented across all clients. Each client map represents objects in the central object and topology databases.

- New IP objects in the database will show up on each map
- Objects will be either managed or unmanaged for all maps. Management is similar for Acknowledge and Unacknowledge.
- Object removal (deleting an object removes it from the database and all maps)

Local Map Features

Local map features contain cosmetic presentation changes, as follows:

- Submap layout, sorting
- Hide/Unhide
- Cut and Paste
- Non application symbols, such as location and other manually created symbols

Tips about Deleting Objects

Here’s some tips about deleting objects you may find helpful.

- If you delete a submap object, all the child objects will be deleted; therefore, those symbols on all client maps are also deleted. If you delete a map, only the map will be deleted, but no objects.
- If you delete the last symbol on any map for that object, the object will also be deleted. If you cut a symbol but exit before doing a paste, the symbol will be deleted. That is, the cut has been converted to a delete symbol.

Time Synchronization between NetView Server and Clients

NetView attempts to keep the system time of the clients and server in synchronization. This is important for maintaining the propagated map actions, such as manage, unmanage, acknowledge, and unacknowledge on each client’s map.
By default, the time is checked when the NetView Console starts and every hour after. You can change the time check by setting the following system variable in the System dialog on the Control Panel:

\[ \text{TIMESYNC\_INTERVAL}=3600 \text{ (seconds)} \]

By default, the tolerance is 1 second. You can change this by setting the following system variable:

\[ \text{TIMESYNC\_VARIANCE}=1 \text{ (seconds)} \]

---

**Client/Server Map Propagation**

The client/server model for the maps consists of a centralized object and topology database, but a localized map database. This means that you can expect all maps to share object and topology properties including object status and configuration attributes. Because the map database is localized, you can customize each client map differently. A set of key map operations are propagated to all maps:

- Manage
- Unmanage
- Acknowledge
- Unacknowledge
- Adding nodes
- Deleting nodes

Only the propagation of adding and deleting nodes is supported. Use the `getservermap.bat` utility to manually distribute customized maps.
Chapter 5. Working with the NetView Program

The NetView program creates and displays windows that contain views of your network topology. This chapter describes the major components of the NetView program and explains how these components work together to help you monitor and manage your network. The following topics are described in this chapter:

- "Learning about Objects"
- "Learning about Symbols" on page 34
- "Learning about Maps" on page 40
- "Learning about Submaps" on page 43
- "Learning about Applications" on page 50
- "Configuring a Graphical Map" on page 53
- "Additional NetView Tools" on page 57

The NetView components are interrelated, and references are made to each in the descriptions of the others.

Learning about 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 properties of the object. Examples of resources represented by objects include:

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

Objects can be created by users or by applications that are integrated with the NetView program. With the NetView program you can add objects and their corresponding symbols to submaps. If objects and symbols are not discovered by the automatic discovery process, you might want to customize your network hierarchy and add objects to submaps by selecting Object..New. For information, see "Add Objects" in the online help. For information about manipulating objects, refer to the Tivoli NetView for Windows Programmer's Guide.

Understanding Object Properties

The properties of each object are contained in fields that make up the object. Each field contains information about the object, such as:

- The object name
- The object IP address
- Whether the object supports SNMP
- The type of hardware the object represents
- The object's status

The entries in these fields define the object and are called object properties. You can display and modify object properties by selecting the object and then selecting Object..Object Properties.

Understanding Capability Fields

An important type of object property is the capability field. A capability field is a Boolean field, which can take the values True and False, and indicates whether the object has a certain characteristic or capability. In the list under "Understanding Object Properties", the third item, Whether the object supports SNMP, is a capability field.
Capability fields govern the actions that you can take when working with an object. When you select an object from a submap, the capability fields of that object help determine which menu options are enabled.

**Finding an Object**
To find an object, such as a node, network, or interface card, either select Edit..Find from the main menu or click on the Find Object icon in the NetView Standard toolbar.

**Managing and Unmanaging Objects**
An object can be managed or unmanaged by selecting Object..Manage or Object..Unmanage. A managed object is monitored by the NetView program for topology, status, and configuration changes. The symbol for a 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.

**Acknowledging and Unacknowledging Objects**
If you know that an object has stopped functioning, but you do not want the NetView program to notify you continuously about this problem, you can select Object..Acknowledge.

When you acknowledge the object, the object turns to the color set for acknowledged symbols and remains in the acknowledged state until you either select the object and unacknowledge it or until the status of the node changes. Unacknowledging an object causes the NetView program to resume normal processing for that object.

**Defining and Managing SmartSets**
You can group objects together in SmartSets. A group of objects is called a SmartSet and is defined by selecting Submap..New SmartSet.. from the main menu, specifying at least one condition, and then clicking the Create SmartSet button.

You can define the following SmartSet types:
- User-defined SmartSets
- Service SmartSets

User-defined SmartSets enable you to define and manage network objects as one item. You can use user-defined SmartSets to set up work assignments for operators or group together systems that support a specific business function such as Order Entry. The NetView program provides default user-defined SmartSets for objects that are commonly grouped together, such as Routers, and for objects that you want to group according to business functions.

You can define a SmartSet to monitor a set of devices closely by selecting Submap+ New SmartSet from the main menu. When you create a SmartSet, you can specify the selection names of the objects or define a rule to be used in creating the SmartSet. If you specify a rule, such as isRouter=True, a SmartSet is created of all objects that fit that description. For example, you can set up a user-defined SmartSet of all critical routers. After defining this SmartSet you can navigate to the resulting submap at any time to get a quick view of the routers.
whose current status is critical. You can also create a SmartSet of all nodes that contain serial interfaces (using the rule hasSerial=True). This SmartSet would be useful in spotting unauthorized PPP lines.

You can customize the attributes available for creating a SmartSet in the Find Simple tab page using the \usr\ov\conf\c\finddialog.conf file. Using this file, you can modify the properties available in the Type and Other drop down lists that appear when you choose Other Properties in the Find by drop down list at the top of this page. Both Type and Other are used to find objects that match particular boolean attributes such as “isComputer” or “isMailServer.” By default, Type displays properties associated with objects of a particular type and Other displays properties associated with SmartSets. To modify the Type or Other drop down list, edit finddialog.conf (following the directions at the top of the file) and then restart the NetView Console.

Service SmartSets are automatically created for you by the nvsniffer program. These SmartSets are based on a single attribute that nvsniffer is responsible for maintaining. For example, nvsniffer might create a service SmartSet named DNSServers based on an attribute, isDNS, being set on the objects that provide the DNS service. In addition to grouping objects based on this common attribute, nvsniffer performs status updates to let you know when a service is no longer available.

Service SmartSets are indicated on the map by a gear symbol inside the folder in the SmartSet icon.

SmartSets are aggregated under a SmartSets icon that is displayed on the Root submap. If you click the SmartSets icon, you see a submenu containing symbols for all the SmartSets that have been defined. On the SmartSets Submap, you can click the individual SmartSet icons to see the objects contained in that SmartSet.

As objects are dynamically updated, they are continually checked against the given SmartSet condition to ensure that the given SmartSet submap displays only those objects that currently match the condition. In addition, when viewing service SmartSets, you can double click the nodes to see the services they provide.

The status displayed for nodes appearing in user-defined SmartSets is based solely on the IP status. In service SmartSets, node status is based on not only the IP status but also whether the given service is currently available. For example, if a node appears as critical in the DNSServers service SmartSet, this means that either the IP status of this node is critical or the DNS Service that this node provides is down. You can double click into the node to see whether it is the service or the network interface card that is down. The service objects that appear at this level can optionally contribute to the IP status. Use Edit -> Properties...Map tab page...Map Application...Properties to specify whether services should contribute to IP status.

The following applications enable you to specify SmartSets instead of specific nodes when defining groups of nodes to act upon: Collect MIB Data, Event Browser and SNMP Options.

The NetView program provides a number of predefined SmartSets such as Routers and Critical Nodes. Double-click the SmartSets icon on the Root submap to display the predefined SmartSets that the NetView program supplies.
NetView supports a SmartSet-based polling mechanism which allows NetView to perform polling based on SmartSet membership. For example, the default polling interval for nodes might be set to 30 minutes, but for members of the Routers SmartSet you can specify a polling interval of five minutes to override the default.

**Quick Refresher Course on Boolean Logic**

When you define SmartSets, 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 SmartSet, “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 SmartSet, “Locate a PC OR a mainframe,” it will find all the PCs and all the mainframes in the network.

When you use SmartSets, 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.

**Displaying the NetView Object Database**

Object property information is stored in the NetView database. If, for example, an object in your network is not showing up on the submap, you can check the object database to determine whether the object is in the database and what its properties are. You can display the contents of the NetView database with the `ovobjprint` command. For more information about this command, refer to the *Tivoli NetView for Windows Programmer’s Reference*.

**Learning about Symbols**

A symbol is the icon used on a submap to represent an object in your network. Different symbol types are used to depict different kinds of network objects. For example, a square symbol type is used to show all computers (PCs, workstations, mainframes, and so forth), and a diamond-shaped symbol type is used to show all connectors (like bridges, gateways and repeaters).

In some cases, a single object can be represented by several symbols. For example, the IP Map application determines the status of a node object based on the operational status of IP interfaces installed on the node. If all of a node’s IP interfaces are down, the IP Map application reflects the node’s status as critical. From the perspective of the IP Map 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 want to create different symbols to represent different states of the one object.
A symbol has several properties that define how the symbol is used in the NetView program. The following sections describe symbol properties.

**Symbol Type**

The symbol type is represented by the outer shape of the symbol, and the symbol subtype is represented by the graphic shown within the shape. For example, the symbol used for a personal computer has a square outer shape and a personal computer icon for its graphic.

Figure 12 shows one of the personal computer symbols used in the NetView submaps.

![Personal Computer Symbol](image)

**Figure 12. Personal Computer Symbol**

You can view all of the symbols provided with the NetView program by selecting Help...Legend from the main menu. If necessary, you can define new symbol types using Symbol Type Registration Files, which are described in the Tivoli NetView for Windows Programmer’s Guide.

**Symbol Location**

A symbol can reside on either the application plane or the user plane of a submap. Symbols created by the user are shown on the user plane, and they have shadows behind them. For more information about submap planes, see “Understanding Submap Planes” on page 47.

**Symbol Behavior**

Symbol behavior defines what happens when you double-click on the symbol or when you click on it and press Enter.

**Explodable Symbols**

When you double-click on an explodable symbol (or click and press Enter), a child submap is opened. An explodable symbol is a symbol that opens to display a lower level of the submap hierarchy.

**Executable Symbols**

When you double-click on an executable symbol (or click and press Enter), the program represented by that symbol is executed. An executable symbol is displayed as a raised button on the submap.

**Web Manageable Symbols**

Some network objects can use web-based home pages for configuration and monitoring. The NetView Console supports this capability, enabling access from either the shortcut menus for the object (Management Page) or the main menu (Tools -> Web Browser -> Management Page). This feature is also available from the Web Client. To enable web based home pages, modify the file `usrlov\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 other page of the Object Properties dialog.

Symbol Label

Each symbol has a label that describes the object represented by the symbol. The label is displayed below the symbol. Because the NetView program 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.

Symbol Status

When the NetView program displays symbols on a submap, the color of each symbol indicates the status of the object or connection that the symbol represents. For example, an object whose symbol is displayed with a green background is functioning normally. A red background means the object or connection is in critical condition. [Table 6] shows the default colors used to display status for both icon symbols and connection symbols.

The NetView program uses a predetermined set of rules that determine the status to display for each symbol on a submap. See “Status Propagation” on page 38 for more information.

Table 6. Default Symbol Status Colors

<table>
<thead>
<tr>
<th>Status</th>
<th>Status Meaning</th>
<th>Default Icon Color</th>
<th>Default Connection Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown</td>
<td>Status cannot be determined.</td>
<td>Blue</td>
<td>Black</td>
</tr>
<tr>
<td>Normal</td>
<td>Normal operational state</td>
<td>Green</td>
<td>Black</td>
</tr>
<tr>
<td>Marginal</td>
<td>Impaired, but still functional</td>
<td>Yellow</td>
<td>Yellow</td>
</tr>
<tr>
<td>Critical</td>
<td>Not functioning</td>
<td>Red</td>
<td>Red</td>
</tr>
<tr>
<td>Unmanaged</td>
<td>Not being monitored. The user has marked this symbol.</td>
<td>Wheat</td>
<td>Black</td>
</tr>
<tr>
<td>Unreachable</td>
<td>Not currently reachable from the management station.</td>
<td>White</td>
<td>Black</td>
</tr>
<tr>
<td>Acknowledged</td>
<td>Not being monitored. The user has marked this symbol.</td>
<td>Dark Green</td>
<td>Black</td>
</tr>
<tr>
<td>User1</td>
<td>A node is down for reconfiguration. This status is cleared when the node again becomes operational.</td>
<td>Pink</td>
<td>Black</td>
</tr>
<tr>
<td>User2</td>
<td>Indicates a failure that cannot be reset by netmon polling. If User2 is associated with a netmon status trap, the netmon status will override User2.</td>
<td>Violet</td>
<td>Black</td>
</tr>
</tbody>
</table>

Understanding Router Fault Isolation

The Router Fault Isolation (RFI) feature reduces the diagnostic cost of locating a network problem, especially when the network becomes partitioned as a result of the failure. RFI simplifies the notification action by issuing only one summary alert identifying the router nearest the fault.
When you use NetView 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 enables you to detect 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 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 occluded and considered unreachable. Without Router Fault Isolation, NetView 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 reachability of the router interfaces connected to that subnet. During the router check, each interface and its subnet are analyzed. A non-responding interface triggers checks of the interface and any connecting routers. Router Fault Isolation generates an event for each Router Down or Router Marginal event. 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>
<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 occluded.)</td>
</tr>
<tr>
<td>Router Unreachable</td>
<td>The network management workstation cannot query the router because it is occluded.</td>
</tr>
<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.
The Router Fault Isolation feature, also known as Event Suppression, is active by default in NetView. 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 in Programmer's Reference for more information. For a detailed explanation of Router Fault Isolation, see the description in /usr/OV/doc/RouterFaultIsolation.htm.

Status Propagation

The NetView program sets the symbol colors on a submap based on the status of objects on its child submaps. This process is called status propagation.

The NetView program can use any of three status rules to determine how status will be propagated (or reported up) from child submaps to the parent submap. The status rules are described in these sections:

- "Default Status Rule"
- "Most Critical Status Rule" on page 39
- "Threshold Status Rule" on page 39

To change the status rule for an existing map, select Edit..Properties from the NetView main menu bar, and then select the Map tab page. Change the status rule in the Propagate Status field. To set the status rule for a new map, select File..New Map.

A special case of status propagation is used in determining node status. The default settings will, for example, draw a node as Critical in one segment and Normal in another segment if the node contains two IP interfaces, one in a Critical state and the other in a Normal state. If the node contains any service objects, the default settings will also consider the state of these service objects when determining the node status. See "Using Node Submaps" on page 52 if you would like to change the defaults used in determining node status.

Status Propagation for Nodes With Multiple Interfaces

Red, green, and yellow are the NetView default colors for status. If a node, such as a router, has multiple interfaces and one interface is down (critical), then the node is red inside the subnet and segment where the critical interface resides. The node is green in all other subnets and segments where the node’s interfaces are up (normal). The node is yellow (marginal) at the IP Internet level, where the compound status of the node is displayed.

Default Status Rule

You can use the default status rule for determining status propagation. See Table 8.

Table 8. NetView Default Status Rule

<table>
<thead>
<tr>
<th>Object Status</th>
<th>Condition of Symbols in the Child Submap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>All symbols are normal, acknowledged, or User1.</td>
</tr>
<tr>
<td>Critical</td>
<td>At least one symbol is critical or User2, and no symbols are normal.</td>
</tr>
<tr>
<td>Marginal</td>
<td>Any of the following:</td>
</tr>
<tr>
<td></td>
<td>• All symbols are marginal.</td>
</tr>
<tr>
<td></td>
<td>• Some symbols are normal and some symbols are marginal.</td>
</tr>
<tr>
<td></td>
<td>• Some symbols are normal, some symbols are marginal, and some symbols are critical.</td>
</tr>
<tr>
<td>Unknown</td>
<td>No symbols with a status of normal, critical, marginal, or unmanaged.</td>
</tr>
</tbody>
</table>
Most Critical Status Rule
You can specify that the status of the most critical symbol in the child submap is propagated to the child submap's symbol in the parent submap. In this case, if any symbol on the child submap is marginal, the symbol on the parent submap is yellow (or marginal). If any symbol on the child submap is critical, the symbol on the parent submap is red (or critical).

Threshold Status Rule
You can specify the percentage of symbols on the child submap that must be marginal or critical before the child submap's symbol on the parent submap is changed to marginal (or yellow). You can specify a second percentage that indicates when the symbol on the parent submap will be changed to critical (or red). These percentages are called the marginal threshold and the critical threshold.

The NetView program calculates each threshold by adding the number of marginal and critical symbols in the child submap. For example, if there are five symbols in the child submap with the marginal threshold set to 40% and the critical threshold set to 80%, status is propagated as follows. If two or three symbols (40% to 60%) are critical or marginal, the parent status is marginal. Figure 13 shows the marginal status propagation. If four or five symbols (80% to 100%) in the child submap are critical or marginal, the parent status is critical. Figure 14 shows the critical status propagation.

You can set thresholds from 0% to 100% and set the critical threshold higher than the marginal threshold.

Figure 13. Marginal Status Is Propagated

Figure 14. Critical Status Is Propagated

If any objects in a submap are User1, their status is ignored unless User1 is the only status in the submap.
Copying, Moving, or Deleting Objects and Symbols

Use the **Edit..Cut**, **Edit..Copy**, **Edit..Paste**, and **Edit..All Submaps** operations to copy, move, or delete symbols on the current submap, or on other submaps in the same map. You might want to perform these operations to organize your network in a more logical manner.

If you have a map open with read-write authorization and one or more objects selected, you can cut or copy symbols of selected objects from one or more submaps in the current map, then paste these symbols onto any submaps (including the same one) for the current map or a different map.

Learning about Maps

A NetView map cannot be viewed on the screen. It is not a picture. Instead, it is an entry in the NetView map database that defines which objects and connections in your network should be monitored as a group. While the NetView program does not display a whole map, it does display each section of a map in a submap. Each submap associated with a map is a picture of the part of your network contained in that submap.

Each map contains a subset of the objects defined in the NetView object database. The object database holds all the definitions of the objects and connections in your network. These objects are represented by symbols that appear on the submaps for displaying each map.

As you are using the NetView program, you can create maps, delete maps, or view any of your existing maps. You can even create several maps and control which applications operate on these maps. Although you can create several maps, you can only open one map at a time.

While you create maps and define their scope, applications dynamically update maps to reflect the current state of your network. So even though you may not display a particular map every day, the NetView program is still monitoring the objects in that map.

Different maps can display different management domains, or they can provide different views of the same domain. For example, the default map created when the NetView program is started shows your entire network, as defined in the object database. If you wanted to pay particular attention to routers, you could create a second map that shows all the routers in your network in one submap. Then you could define your new map as the default map and its router submap as the home, or default, submap that is displayed when the NetView program is started. The map of your entire network is still stored in the map database, and you can view it whenever you want.

Submaps are described in "Learning about Submaps" on page 43. You can create maps and submaps that show your network in a way that helps you manage it more efficiently. See "Create a Map" and "Create a Submap" in the online help for additional information.

Each map in the database contains the following information:

**Name**  The name of the map.

**Root submap**  The highest-level submap of the map. The root submap cannot be deleted.
Home subplot
The subplot that is displayed in the initial subplot window when you open the map. The root subplot is the default home subplot.

Layout algorithm for root subplot
The layout algorithm used for the root subplot. The default is row/column.

Propagate status scheme
The propagate status scheme that applies to the entire map. This scheme determines how the NetView program propagates status from symbols in child submaps to the symbol of the parent object. See "Status Propagation" on page 38.

Application Properties
Properties for any configurable map applications that are available on your system for that 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 client/server or distributed network environment, the NetView daemons run on the server, and the NetView console runs on the client. The client obtains event and topology status information from the server. A client/server environment enables you to distribute 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 NetView console maintains the map database.

Some client/server facts:

<table>
<thead>
<tr>
<th>What</th>
<th>Client</th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetView Console</td>
<td>Runs on clients</td>
<td></td>
</tr>
<tr>
<td>NetView Daemons</td>
<td>Run on server</td>
<td></td>
</tr>
<tr>
<td>Map Database</td>
<td>Resides on client</td>
<td></td>
</tr>
<tr>
<td>Object Database</td>
<td>Resides on server</td>
<td></td>
</tr>
<tr>
<td>Topology Database</td>
<td>Resides on server</td>
<td></td>
</tr>
<tr>
<td>Event and Topology information</td>
<td>Provides to client</td>
<td></td>
</tr>
<tr>
<td>Network Discovery and Status polling</td>
<td></td>
<td>Performed by server</td>
</tr>
</tbody>
</table>

Network and Subnetwork Aliases
You can assign names and aliases to specific networks and subnetworks using the %systemroot%\system32\drivers\etc\networks file. Place an entry in the file for each network or subnetwork you want to assign an alias. Each entry in the file represents one network. When the NetView program discovers the network, the NetView program displays the network on the map using the name from this file as the label.

The format of the file is:
Customizing Maps

You can customize maps to meet the needs of individual users or to display a network in a way that helps you manage it more efficiently. Customizing a map enables you to:

- Allocate responsibility for managing your network among several people. For example, network administrators, who are 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 example, data traffic or performance monitoring.
- Create a map that is a projection of a network 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 object database. Select Edit..Properties to customize map information.

Understanding the Open Map

The open map is the active map that is currently displaying a hierarchy of submaps. This map can be updated by applications and users. Each user can open only one map at a time during a NetView session.

You can choose to 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 specific areas of responsibility on your network. In large or complex networks, it can become impractical for one map to display all information about systems on your network. You can create maps that focus on a specific set of system or node 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 properties, 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 operating system 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. It is useful to keep a copy of a map before making changes to your network that can result in an uncertain outcome.

- 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 use different background pictures for the different submap segments.

- Map Applications

You might want to integrate an application with the NetView program to provide specific information about systems on your network. The type of information depends on the application. Applications can monitor and control different aspects of the objects on your network. For example:
- Nodes on your network with an IP address
- Remote file servers and their clients
- Specific systems on your network

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. Snapshots are read only and cannot be updated by applications.

Use snapshots to document your network or to keep a record of your network’s status. It is useful to take snapshots before making major configuration changes.

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. Highlighted objects are specific to the map or snapshot.

To create a map snapshot, select File..Save As from the main menu. You then select Snapshot on the Save As dialog box. You can list all the snapshots you have saved by selecting Window..Maps or by selecting File..Open. For more information about map snapshots, see “Taking Snapshots of a Map” on page 99 and the online help.

Learning about Submaps

A submap is a collection of related symbols that are displayed in a single window. A submap provides a view into a map. Each submap displays a different perspective of the information in the map. Typically, submaps are organized in a hierarchy that enables you to view 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.

Both applications and users can create submaps. An application can create and open submaps that display symbols of objects that the application manages. After submaps are created, you can view them by selecting Window..Submaps from the menu bar.

Usually, users navigate through submaps by double-clicking the mouse on explodable symbols or by clicking once on a symbol and pressing Enter. Double-clicking on an explodable symbol causes a subnet map to be displayed. The
object associated with the explodable symbol is called the parent object. The submap that is displayed by double-clicking on the symbol associated with the parent object is called a child submap.

Each time you double-click to display a new submap, it replaces the existing submap in the submap window. To open a second submap in a new window (instead of replacing the existing submap), select Submap..Open in New Window or press Shift and double-click on the submap symbol. You can return to a previous submap in any of the following ways:

- Use the submap context menu to select the parent of the current submap.
- Use the Navigation Tree to select the submap you want to view.
- Double-click on the submap at the bottom of the submap stack to the left of the current submap display.
- Select View..Parent from the main menu.
- Press the Ctrl key and the Up Arrow cursor key.

The following sections describe ways you can work with submaps to manage your network. For more information about any of these tasks, use the online help while performing the task.

Working with Submaps

Submaps enable you to do the following:

- 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.

Table 9 describes submap properties.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the submap, assigned when the submap was created. Each name must be unique within the same map.</td>
</tr>
<tr>
<td>Parent Object</td>
<td>The object is the parent object of a submap. Because several symbols from different submaps can represent the same object, you can navigate to a child submap from several submaps. Each of the symbols representing the object open a submap. However, a given submap might not have a parent object.</td>
</tr>
<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. After the algorithm is set, it cannot be changed for the submap.</td>
</tr>
<tr>
<td>Presentation</td>
<td>The presentation format used by the NetView program to display the submap. The presentation can be either scaling or zooming.</td>
</tr>
<tr>
<td>Background</td>
<td>The picture displayed on the background of the submap plane to customize the appearance of the submap.</td>
</tr>
<tr>
<td>Comments</td>
<td>The comments about the submap.</td>
</tr>
</tbody>
</table>

You can create, delete, and modify the properties of submaps in the open map by selecting Edit..Properties. For example, you might want to change a submap’s presentation, background picture, or its parent. Or, you might want to create a submap to display a detailed view of systems on your network. You can continue to create and view 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. For more information about creating, deleting, and modifying submaps and submap properties, see “Submaps” in the online help.

You can create objects and connections on submaps. If objects and symbols are not discovered by the automatic discovery process, you might want to customize your network hierarchy and add objects to submaps by selecting **Object..New**. For information on how users can create objects, see "Add Objects" in the online help. For information on how applications can manipulate objects, refer to the *Tivoli NetView for Windows Programmer’s Guide*.

You can add a connection symbol between any two object symbols or between an object symbol and a backbone on a submap by selecting **Objects..Connect Objects**.

**Using the Root Submap**
The NetView program creates a **root submap** that provides a standard, top-level submap on which you can display the symbols that represent multiple networks or subnetworks. The root submap enables you to place multiple networks 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 objects. You can select one of these objects and display the highest level of a submap hierarchy.

When a map is created, the IP Map application is enabled by default. As soon as the map is created, the netmon daemon begins discovering IP-addressable objects across the network. Then the IP Map application builds the map to reflect the discovered database objects. During the initial discovery process, the IP Map application places a single internet symbol on the root submap. The internet symbol graphically represents all IP-addressable objects on your network. After you create the root submap, you cannot delete it, nor can you delete the internet symbol.

**Using the Home Submap**
Each map has a submap designated as the **home submap**. The home submap is the first submap that the NetView program displays when it opens a map.

You can assign any submap in the map as the home submap by selecting **Edit..Properties**. For example, if you constantly find yourself clicking your way down to a particular segment in the network, you might want to designate that submap segment as your home submap.

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. You cannot delete the root submap.

**Creating Child Submaps and Independent Submaps**
In the NetView program, 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 either a detailed view or the contents of an object called the parent object on the map. To create a child submap, double-click on an
explodable parent symbol that has no child submap. For more information on creating submaps, see “Creating and Customizing Submaps” in the online help.

An independent submap has no parent object or parent submap. To create an independent submap, select **Submap..New**. To display an independent submap, select **Submap..Open** or **Window..Submaps**, or select the independent submap from the Navigation Tree window. For more information, see “Creating an Independent Submap” in the online help.

**Understanding Submap Presentation**

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

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

Zooming enables you to display a close-up or far-away view of the submap. Scroll bars appear when only a portion of the submap fits into the viewing area. You can set the zoom factor to determine the extent to which you zoom into the view of the submap. The default zoom factor is **one**.

To change from scaling to zooming, select **View..Zoom Map**. To change back to scaling, select **View..Scale Map**. To customize your view of the submap while in zooming mode, select **View..Zoom In** or **View..Zoom Out**.

**Understanding Submap Layouts**

The way the NetView program arranges symbols on a submap is called the submap layout. The method for arranging symbols on the submap is called the layout algorithm. You can automatically place symbols on a submap as determined by the layout algorithm, or you can place them manually.

**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 10** lists the available layout algorithms. You can also view the layout algorithms online by selecting **File..New Map**.

**Table 10. 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 interconnected 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 of nodes around a center symbol.</td>
</tr>
<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>None</td>
<td>Symbols are arranged by the user or are left in the New Objects 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 afterward. If an application creates a submap, it can specify a layout algorithm. If no layout algorithm is specified when the submap is created, a default layout algorithm, based on the symbol type of the parent object, is assigned.

By default, token ring submaps are drawn as a bus with rounded turns while Ethernet Submaps are drawn as a bus with square turns. Large token–ring segments can be easily viewed.

**Using Automatic Layout**
The NetView default is to automatically lay out new objects on all submaps. You can reset automatic layout for a single submap, or for all submaps. To reset automatic layout for a single submap, select **Submap..Automatic Layout**. To reset automatic layout for all submaps, select **Edit..Properties**.

**Note:** Turning off automatic layout does not enable you to change the layout algorithm for a map.

**Using Refresh Layout**
Select **Submap..Refresh Layout** to arrange the symbols in a submap according to the assigned layout algorithm. This refresh works regardless of whether automatic layout is enabled or disabled.

You might want to refresh the layout of a submap if automatic layout is turned off, and you want to add objects from the New Objects area to the submap. Also, if you have manually moved symbols on the submap and now want to see the default submap layout, you can use Refresh Layout to return the submap to its default layout.

**Using the New Objects Area**
A New Objects 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. Symbols in the New Objects area are shown without their connections.

You can drag symbols from the New Objects area (using mouse button 1) and place them in the associated submap.

**Understanding Submap Planes**

Submaps contain the following three **planes** or layers:

- Background plane
- Application plane
- User plane

The **background plane** provides the background against which symbols are viewed. You can add background pictures in the background plane to provide a context for viewing symbols in your submap presentation. Select **Edit..Properties..Submap** to add a background picture. Symbols on the application plane and the user plane are displayed on top of the picture.

Background pictures 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
The background can be different for each submap. For more information on adding background pictures, see "Adding or Changing a Submap Background" in the online help.

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 NetView program distinguishes symbols on the user plane by providing shadows for symbols to make them appear raised above the submap background. Select Help..Legend..Other to see the symbol for a user object.

**Metaconnection Submap**

You can create multiple connections between two objects by selecting Objects..Connect..Object. When you do, the NetView program displays a metaconnection symbol on the submap containing the two objects. When you later double-click on the metaconnection symbol, a submap is displayed that shows each of the connections. This submap is the metaconnection submap. Figure 15 on page 49 shows an example of a metaconnection submap.

The NetView program allows you to create an unlimited number of connections either between two submap symbols that represent network objects or between an object symbol and a backbone. The NetView program automatically creates a metaconnection submap when you or an application adds a second connection between two symbols or between a symbol and a backbone. It also automatically adds to the metaconnection submap each new connection that you create between the two symbols. The metaconnection submap displays the status of each connection between the two symbols.

If the metaconnection submap also has a child submap, you can access the child submap by selecting Window..Submaps. The submap view contains a list of the submaps and the operations that are specific to submaps.
A metaconnection submap has the following properties:

- Displays all the connections represented by the metaconnection symbol.
- Displays a row/column layout unless the connections are between an object and a backbone. If the connections are between an object and a backbone, the picture shows all the symbols representing the objects connected to one backbone.
- Displays in the metaconnection submap the two endpoints for each connection in the submap (unless the connection is between an object and a backbone).

**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 any of the objects in the metaconnection submap. Therefore, the metaconnection submap can be a parent of other regular submaps.
- You can select 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 view propagated status for connected icon symbols. The metaconnection symbol displays the propagation status of the multiple connections in the metaconnection submap. Unconnected objects in the metaconnection submap also contribute to propagation status. However, the connected icon symbols in a metaconnection submap do not propagate their status. Their status is maintained by the symbols in the parent submap above the metaconnection submap.

Learning about Applications

In this book, the term application refers to any program with which you interact through the NetView program. Applications can provide the following functions:
• Process user requests
• Create or delete objects, symbols, and submaps
• Change the contents of maps
• Provide special display functions
• Provide additional network management capabilities.

You can write applications and integrate them with the NetView program. Refer to the Tivoli NetView for Windows Programmer’s Guide for more information about writing applications.

Understanding the IP Map Application

The IP Map application is the primary application used by the NetView program. When you start the NetView program, the IP Map application is automatically started, and it supports processing of submaps and symbols. The IP Map application creates an object for each IP-addressable node discovered in your network. It also creates and displays symbols on your network map that represent those map objects.

If you delete the symbol for an IP object from one map, but the symbol still exists in another map, the object is handled in the following ways:
• The object is not redrawn on the map from which it was deleted.
• The object still exists in other maps.
• The object is not deleted from the map database.

An IP object is deleted from the map database only when its corresponding symbols have been deleted from all IP maps in which they occur. However, the object might subsequently be rediscovered by the netmon daemon and redrawn on the map. If you do not want an object to be managed or to appear on the map:
• Select Object..Unmanage to unmanage the object from all submaps in the open map.
• Select Edit..Hide to hide the object.

When you do not want an object to appear on your submaps and you do want the NetView program to continue gathering information about that object and managing its status, you can just hide the object.

The IP Map application uses information from the netmon and ovtopmd daemons to discover and update the map. The netmon daemon discovers resources in your network, and the ovtopmd daemon continually accesses stored semantic information about IP networks, segments, nodes, and interfaces in the map database. As information in the map database changes, and as events take place with systems or connections in the network, the IP Map application updates your network map. Thus, if the IP Map application fails, the NetView representation of
your network is not accurate. For more information about daemons, refer to the online help, the *Tivoli NetView for Windows Programmer’s Guide* or the *Tivoli NetView for Windows Programmer’s Reference*.

**Understanding Map Updates**

When a map that uses the IP Map application is opened, the IP Map application starts updating the map with all the network changes that have occurred since the map was last opened. While the IP Map application is updating (or synchronizing) the map, the NetView program displays the *Updating Map* message on the status line of all displayed submaps of the open map. During this update phase, the IP Map application requests information about changes to the IP topology database since the map was last open. This information is constantly updated by the ovtopmd daemon. If new objects are discovered, the IP Map application brings the map up-to-date by drawing new symbols.

While the IP Map application is updating a map, the following limitations exist:

- You cannot delete symbols, objects, or submaps.
- You cannot see updates to the map for symbols and objects you add.
- The Verify button on the IP Map Properties dialog box does not respond while IP Map is updating a map.
- The Object..Manage and Object..Unmanage operations do not take effect immediately. These actions are queued for later processing if you request them during map synchronization.
- The IP Map application may not appear in the Other tab page of the Object Wizard – Step 2 of 2 dialog (accessed by selecting *Object..New*) or in the Object Properties dialog.

When the map update is completed, the IP Map application resumes full operation and the *Updating Map* message is no longer displayed.

**Understanding the IP Map Application Submap Hierarchy**

When a map is created, the IP Map application creates a hierarchy of submaps that display your network in increasing detail. From the internet symbol on the root submap, you can open and display the highest-level internet submap. You can reorganize and expand the hierarchy of submaps. The IP Map application initially creates the following submap hierarchy:

- Internet submap
- Network submaps
- Segment submaps
- Node submaps

The IP Map application manages only objects that exist within this hierarchy of submaps. The objects of symbols added to the root submap, or added outside the realm of the IP submap hierarchy, is not managed by the IP Map application.

**Using the Internet Submap**

The internet submap shows the logical partitioning of IP networks and subnetworks connected by gateways on the graphical map.

You can create several partitioned internet submaps to distribute resources across network and subnetwork submaps. Creating partitions provides you with more control over the organization of your network map, and enables you to partition your
map geographically, or to show administrative domains. The IP Map application always places discovered IP-addressable gateways and networks on the original, highest-level internet submap.

Select Edit..Cut and Edit..Paste to move these objects onto partitioned internet submaps. For more information, see "Configuring a Graphical Map" on page 53.

Using Network Submaps
The network submap represents the physical topology of a network at the level of network segments. The IP Map application can discover and display objects such as IP-addressable segments, gateways (routers), bridges, and the connections between them on the network submap. You can add objects to the network submap.

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

A network submap can contain segment symbols in the network class and symbols in the connector class.

Using 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, with varying degrees of intelligence, 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 IP Map application can discover and display the following segment topologies:

- **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)

Using Node Submaps
The node submap displays symbols that represent the components of a node in a row/column layout. The IP Map application places interfaces on the node submap. When an interface object is added to a node submap, the IP Map application places a connection symbol that represents the interface object in higher-level submaps. In addition, if the nvsniffer program determines that the node contains services, the Collmap application places services on the node submap.

The default propagation of objects within node submaps to the associated nodes in the IP topology is based on both the IP interface objects as well as the service objects. You can, however, customize this propagation in certain ways. Use Edit → Properties/Map- tab- page/MapApplication/Properties to specify such things as:
• Whether services should contribute to IP status
• Whether a node should be displayed with the same status in all submaps (by not associating a particular IP interface object with the particular segment the node appears on). Use **Exclusively Set the Status for Nodes** to change this.

## Configuring a Graphical Map

This section helps you configure a map to represent your network. You can do this manually, or you can create a custom configuration file to automatically customize your network layout.

### Map Regions

The set of objects on a particular map is called the **map region**. You can customize a different map region for each map.

One way to customize the map region is by geography. You can create different submaps for different parts of the country. Each map would show objects based on where they are physically located in the country. You could also create map regions based on specific types of nodes, such as gateways and routers.

Using different map regions, you can divide responsibility for managing systems on your network. For example, if you have regions that focus on physical areas of your network, network administrators expert in managing routers or gateways can open a map that is configured to help manage those devices.

### Creating a Map Region

To create a map region, first select **Object..New** or press the **Insert** key. Add one of the objects in **Table 11**.

**Table 11. Container Objects**

<table>
<thead>
<tr>
<th>Symbol Subclass of Object</th>
<th>Symbol Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet</td>
<td>Network</td>
</tr>
<tr>
<td>Site, Site2, Site3</td>
<td>Location</td>
</tr>
<tr>
<td>Room, Room2</td>
<td>Location</td>
</tr>
<tr>
<td>City</td>
<td>Location</td>
</tr>
<tr>
<td>State</td>
<td>Location</td>
</tr>
<tr>
<td>U.S.A.</td>
<td>Location</td>
</tr>
<tr>
<td>Africa, Asia, Australia, Europe, N. America, S. America</td>
<td>Location</td>
</tr>
</tbody>
</table>

These objects are called **container objects**. Container objects have a symbol type of either the internet symbol in the network class or any of the symbols in the location class. Keep adding container objects until you have an object for each site in your network that you want to see in a map region.

The IP Map application manages container objects just as it manages the original objects on the internet submap. For more information about creating objects, see “Adding Objects and Symbols” in the online help.

If you want to create map regions that show types of network devices, you can create new objects for these device types by selecting **Object..New**. If the objects already exist on other submaps, select **Edit..Copy** and **Edit..Paste** to copy the symbols from the other submaps.
The IP Map application automatically recognizes any new symbols on a submap if they are symbol types provided by the NetView program. Table 12 shows examples of these NetView symbol types. To see all of the NetView symbol types, select Help..Legend.

**Table 12. Examples of NetView Symbol Types**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gateway</td>
<td>Connector</td>
</tr>
<tr>
<td>Audio card</td>
<td>Cards</td>
</tr>
<tr>
<td>IP network</td>
<td>Network</td>
</tr>
<tr>
<td>Internet</td>
<td>Network</td>
</tr>
<tr>
<td>Workstation</td>
<td>Computer</td>
</tr>
<tr>
<td>Application</td>
<td>Client</td>
</tr>
</tbody>
</table>

When you cut these symbols from the original internet submap and paste them on a container submap, all submaps, objects, symbols, and connections that are contained under the cut-and-pasted symbol remain intact. Symbol behavior and symbol status are also maintained. If the symbol you paste on a map opens a child submap, then the new symbol opens a submap. In addition to the hierarchical relationship, all the topological relationships between symbols are maintained.

**Automating Internet Submap Customization**

You may 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. You can automate this level of customization in your maps using the location file `\usr\ov\conf\location.conf`. This file, if present, determines how the networks and gateways are laid out on your map. It contains locations and the networks and gateways that should be placed in those locations. Locations are based on networks and gateways; only networks and gateways are placed under a location symbol. All other symbols, such as segments, are placed under the network to which they belong. Locations cannot be used to divide a subnet.

After creating or modifying the location.conf file, you must either create a new map or regenerate the default map in order for the location.conf file to take effect for networks and gateways already existing in your map. As new networks and gateways are discovered, they will be placed as specified in the location.conf file.

To create a new map, select **File→New Map** from the NetView Console. Use Server Setup to clear the map databases and regenerate the default map.

Refer to the sample location.conf file in the `\usr\OV\conf` directory for the syntax of the file. Invalid entries will be documented in the nv.log file and will be ignored.

To delete the current maps and restart map generation, use the **Clear Databases** option in the **Databases** page available from the **Server Setup** application.

Restrictions: If the `/usr/OV/conf/location.conf` file is present, it will be used.

Syntax errors in the location.conf file will be documented in the nv.log file.

**NetView Entries**

Network entries have the following format:

Name AddressPattern LocationIcon <ParentName>
Where:

**Name**  Name for the location symbol displayed on the map.

**AddressPattern**
Networks whose network addresses match this pattern will be placed under the location symbol. See the location.conf file for details on the syntax of this pattern.

**LocationIcon**
Type of location icon to be used. Possible values include: Site, Room, and City. Refer to the Help -> Legend menu item from the NetView Console for a list of all possible values for the Location icon.

**ParentName**
Optionally, a previously defined “parent” location under which this location symbol should be placed. Using the “parent location” allows networks to be nested. Forward references are not valid.

**Auto-placement of Routers:** A router is placed in the lowest nested location that includes all of the networks to which the router is connected as follows:

- If all of the router’s networks are inside one location, the router is displayed in this location connected to all the networks.
- If the router’s networks are in more than one location, the router appears in the first location (up the parent chain), which includes all of the locations for these networks. The router is displayed in the parent location connected to one or more child locations.
- If the router’s networks have no common location in the parent chain, or if one of the networks is displayed in the IP Internet submap, the router is displayed in the top level IP Internet submap.
- If the router is discovered as unmanaged, it is displayed in the top level IP Internet submap. In this case, the interfaces have not been discovered.

**Location File Examples**
The following are examples of location files.

**Example 1:** This location file creates a series of location symbols representing smaller Company ABC offices. All machines in London have addresses in the range 146.84.223.* - 146.84.226.* (specified as 146.84.223-226). These machines and their associated segments and subnets would be placed under a location symbol called London which would live at the IP Internet level of the map. Each location symbol uses the “Location:Site” icon. Router router1 is placed under the London location.

```
scdev  146.84.211-214  Site
sanfrancisco  146.84.218  Site
switzerland  146.84.219  Site
london  146.84.223-226  Site
london  router1.domain.com  Site
```

**Example 2:** This location file creates a nested series of location symbols under a location called SmallOffices. Only the SmallOffices location symbol appears at the IP Internet level, the other location symbols are under SmallOffices. Note that the AddressPattern field for SmallOffices is zero. This implies that no networks should exist under this symbol, just other location symbols. There are two entries for the SuiteB location. Networks that match either of these two address patterns will be added under the SuiteB location symbol.
Example of a Customized Internet Submap

On the internet submap, you can place a background of the world and add symbols that represent container objects of countries or continents. Each container object opens into its own submap. Then, you can cut symbols that represent your resources in a particular country from the internet submap and paste them to any of the container submaps.

Figure 16 shows an internet submap in which four container objects were added. The symbols of the container objects represent Asia, the United States, South America, and Europe. In addition, all symbols of objects discovered by the IP Map application have been cut from the internet submap and placed in lower-level container submaps.

![Example of a Customized Internet Submap](image)

Example of a Customized Submap

Figure 17 on page 57 shows the United States submap. Four container objects have been created in this submap. From this submap, you can double-click on any of the symbols and open a child submap to display other container objects or symbols that represent IP networks and gateways.
Understanding the xxmap Application

The xxmap application is used by the NetView program to manage open topology networks. These are networks that use a protocol other than TCP/IP. For example, the DECnet protocol uses the xxmap application.

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. For more information on the gtmd daemon, refer to the Tivoli NetView for Windows Programmer’s Reference.

The xxmap application supports user changes to the map. Using NetView windows, you can add and delete symbols. However, added symbols are not verified or stored in the gtmd database. These symbols exist only in the user plane.

Additional NetView Tools

To customize the NetView program, the following tools are available.

NVDBTools

Four tools have been added to modify and query NetView’s object database from the command line in order to vastly reduce the effort to customize the database and produce reports. It also allows integration with Tivoli Inventory to collect and retain hardware and software information across the enterprise.

Each utility is in \usr\ov\bin. For more details on each of these tools, see the respective Readme files in \usr\ov\doc.
NVDBImport.exe

This utility can be used to import object field values into the NetView object database from a file. For example, if you want to set a new field called isSite2 to TRUE for the list of nodes at Site 2 and create a SmartSet for them, perform the following steps:

1. Create a file with the list of nodes.
2. Create the field isSite2 with the value TRUE.
3. Run the NVDBImport.exe utility to update the database.

If you need to recreate the database, you can quickly restore your customization. If you already have a customized database, use NVDBFormat to generate this file for you.

Several sample template files are included.

NVDBExport.exe

This utility exports the NetView object database into an SQL database for either Microsoft Access or Microsoft SQL Server. You can either export the entire database into a set of tables, or use the configuration file to customize the table formats and populate them with entries based on rules. For example, you can create a table containing just the critical nodes and relevant fields, such as IP Hostname and sysContact. Then use an SQL report writer to generate reports.

NVDBFormat.exe

This utility can be used to generate custom text or HTML reports on NetView’s object database complete with header and footers including some aggregate information. It can also generate a file of your customizations that subsequently can be used by NVDBImport to update a new database.

Several sample template files are included.

NVDBToMIF.exe

The purpose of this tool is to integrate the power of the NetView program and its network awareness with the scope of Tivoli Inventory to collect and retain hardware and software information across an enterprise.

This tool exports the object database fields for a specific object in the NetView database or for all objects. It then creates the following files in the user’s indicated output directory:

- netview_export.mif
- netview_<DBTYPE>_createtables.sql
- netview_<DBTYPE>_createviews.sql

The SQL files can be used by a Tivoli Inventory user to prepare tables and views within Inventory’s RDBMS. These tables will be used to store the data that is presented in the netview_export.mif file. See the Desktop Management Interface specification for more information about the syntax and content of MIF files.

Using Wake on LAN

This tool can be used to activate machines that are compliant with the “Wired for Management” Wake on LAN protocol. This includes special software and hardware on the target platform. This feature is only available on the Intel platform.
You can run this tool either by selecting **Tools** → **Wired for Mgt** → **Wake on LAN** from the NetView Console or from the command line (**nvwakeup.exe**).

**Desktop Management Interface**

Desktop Management Interface (DMI) is a technology by which PCs can be more manageable by providing extensive information on their hardware and software resources. This feature is not available on the Alpha platform in this release.

From the NetView Console, select **Tools** → **DMI**... to:

- Dynamically query DMI nodes for DMI information relative to specific groups with the Baseline Wired for Management specification.
- Dynamically query DMI nodes for specific group information and then store that information within the native NetView object database. This information can be used to diagnose a machine that is in a critical state.
Chapter 6. Working with Events and Event Filters

To help you manage a network effectively, the NetView program must receive information about changes that affect objects in the network. Traps generated by agents that monitor network objects convey this information to the NetView program. A trap is a message sent from an SNMP agent to an SNMP manager without a specific request from the SNMP manager.

Agents send traps to the SNMP manager to indicate that a particular condition exists on the agent system, such as the occurrence of an error. In addition, the SNMP manager generates traps when it detects status changes or other unusual conditions while polling network objects.

The NetView program provides the event browser for viewing traps, which are considered events by the NetView program. The Event Browser uses a Structured Query Language (SQL) database for accessing and storing events.

Every event and trap is stored in the database and can be displayed in the Event Browser window. However, large networks with many objects and agents can generate so many traps and events that you may want to create event filters. Event filters allow you to specify the kinds of events and traps you want to see in the Event Browser window. Filtering events enables you to keep your attention focused on only those events and traps that are important to you in managing your network.

This chapter contains the following topics:
- "Understanding Events"
- "Displaying Events" on page 63
- "Working with Displayed Events" on page 64
- "Filtering Events for Display" on page 64
- "Displaying Event History" on page 65
- "Configuring Events and Traps" on page 65

Understanding Events

The NetView program uses the following types of events:

**Map events**

Notifications issued when a user or application does something that affects the status of the current map or of the 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**

Messages sent by an agent to one or more managers to tell them an event has occurred that affects 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, even though the agent is not displayed on your map.

**NetView**

Internal events are messages generated by NetView applications.
When Events Are Generated

Events are generated when at least one of the following conditions occur:

- A threshold limit set through the Tools..MIB..Collect Data option was exceeded.
- The network topology changed; for example, an object or interface was added or deleted. This does not include the objects or interfaces that you add to the user plane of the submap.
- An informational message or an error indicating inconsistent or unexpected behavior.
- An ICMP echo request detected that an object’s status changed or that an interface stopped.
- A node’s configuration changed.
- An SNMP trap was received from a managed node.

Where Events Are Stored

All events and traps received by the NetView program are stored in the Events SQL database. You can filter events for display from this 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 NetView program:

- 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 sends traps to the manager system defined in 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 NetView Internal Events

The NetView program’s internally generated events 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 management system running the NetView program.

To look at a list of the NetView program’s internally generated events, enter the \usr\ov\bin\event -l command from a command prompt window.

Information Displayed by Events

Events provide the following information:

- **Description**: The description of the trap
- **Node**: The node that originated the trap
Displaying Events

The Event Browser displays all the events you have filtered for display, as they occur in your network. Some of these events are forwarded to the NetView program from the attached network, and some are internal events generated by NetView applications, such as netmon.

Starting the Event Browser

To start the Event Browser, use one of the following methods:

- Select Monitor..Events from the main menu.
  
  You can display events for all nodes or a selected node. As new events occur in the network, they are displayed in the Event Browser if they match the filter you select.

- Click the Event Browser icon
  
  When you click on the Event Browser icon in the NetView Application toolbar, you display the events for the selected node.

Figure 18 displays the Event Browser.

If you want to change how the NetView program starts, see “Customizing the NetView Console” on page 24.
Customizing the Event Display

The Event Browser displays the description, node, enterprise, trap, severity, category, and source in column format for an event. You can customize the display to show only the columns you need. For example, you can hide the columns you do not want displayed and resize the remaining columns for a better view of their information. Select Options..Columns... to hide or display columns.

Pausing the Event Display

As new events occur in the network, the Event Browser window is updated. The oldest event in the list is scrolled off the top of the list when a new event occurs, and the new event is added to the bottom of the list.

New events may occur before you have finished analyzing the existing events. To make sure the events you are working with are not scrolled, stop the display by selecting View..Pause Display.

Selecting View..Pause Display just stops the event display, it has no impact on the events database. The Event Browser stores events in the events database and reads events from the database for display. When you resume the display, the Event Browser reads the events database and updates the display.

Working with Displayed Events

The Event Browser displays all events filtered for display that are received after it is started. By default, events are displayed in the order in which they are received. The Event Browser lets you perform operations on events:

- Show Node on Map: Click the Show Node on Map icon or select Tools..Show Node on Map from the menu to show the object for which the event was generated. The NetView program displays the submap containing that node and highlights the node. When the node is highlighted, its title is shown in reverse video. The Show Node on Map selection is only available when one event is selected.
- Graph Event Traffic: Select Tools..Graph Event Traffic to create a graph of the overall event rate over time. This can be useful to spot event trends and patterns.
- Generate Statistics: Select View..Statistics to display a breakdown by severity of all events that passed through a filter.
- Add Notes to Events: Select View..Event Detail to attach notes to events. You can use this selection to construct work lists by adding notes indicating that network objects need attention.

Filtering Events for Display

You can view or change filters by selecting Filter..Set. For example, you might not want to see an event every time a workstation is powered off for the day. Or, if several operators are monitoring the network, you might want to create different event filters for each system running the NetView program, so each operator can concentrate on a specific type of network problem or a specific group of devices in the network.

Filtering events involves setting up criteria that an event must meet before it can be displayed or sent to another application. The filters are then stored in the \usr\ov\filters default directory or in a directory that you create. To access filters, the NetView program must know the location of the directory containing the filter files.
To set a filter, select one of the predefined, custom filters from the drop down list. When you create your own custom filter and save the filter, it is automatically added to the drop-down list under the name you give it.

---

### Predefined Filters in the Event Browser

The event browser now supplies predefined filters. You can select a predefined filter from a drop-down list in the event browser toolbar. The predefined filters are:

- **Recent Critical Events**
  - Displays critical events received within the last hour.

- **Topology Events**
  - Displays events with the network topology category.

- **All Events**
  - Displays all events except for those with the log only category.

- **Error Events**
  - Displays events with the error category.

- **Threshold Events**
  - Displays events with the threshold category.

- **Status Events**
  - Displays events with the status category.

- **Configuration Events**
  - Displays events with the node configuration category.

- **Application Alert Events**
  - Displays events with the application alert category.

- **Events with Owners**
  - Displays events with owners.

- **Events with Notes**
  - Displays events with notes.

- **Custom Filter**
  - Does not use a predefined filter.

When you select a filter, the event browser loads the predefined filter and refreshes the event browser to display the events that match the filter.

---

### Displaying Event History

Each event received by the management system is stored in the Events SQL database. This database is stored in `\usr\ov\databases`. Select **Filter..Event History** to display events from this database.

---

### Configuring Events and Traps

Each enterprise has a MIB that describes operations that can be performed on the devices of that enterprise. Enterprises can specify traps that they expect to receive from agents that support their MIBs. You can configure the traps supplied with the MIB to provide more specific information about the status of network objects using the `mib2trap` utility (refer to the *Tivoli NetView for Windows Programmer’s Reference* for more details).

Event configuration offers the following advantages:
You can format a trap to display information that is meaningful to you.

You can associate an action with a trap by specifying the commands that are to be executed when the management system receives an event. When you specify commands, you can automate some fault management procedures and control the amount of event information to be displayed.

You can associate severity with different events. These severity levels are displayed in the event cards. You can search on events by severity level.

To configure traps and events, select **Options..Trap Settings** on the NetView main menu, or select **Options..Trap Settings** from the Event Browser menu. You can specify, for example, that a certain trap prompts trapd to execute a command when an event associated with that trap is received.

To configure traps, the following conditions must be true:

- The node for which you want to configure an event must support SNMP.
- 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 to find out about the enterprise-specific events included with their product.

When an event is configured, it is added to the \usr\ov\conf\trapd.conf file. When this event is received from an agent, the information in the \usr\ov\conf\trapd.conf file is used to format the trap. In addition, the Event Browser reads the event information from the \usr\ov\conf\trapd.conf file and formats it for display.

**Actions on Events**

NetView enables you to specify actions that are automatically performed when a specific event occurs using **Options..Trap Settings**. This feature provides quick notification for events that are critical to your environment. The types of actions you can configure include the following:

- Display an error dialog using **nvecho**. Refer to the *Programmer's Reference* for information on the nvecho command.
- Send an e-mail message using **nvmail**. Refer to the *Programmer's Reference* for more information.
- Send a page using **nvpage**. Refer to the "Paging Support" on page 66 later in this chapter for more information.

You can also configure any action that you can run in a batch file. You must specify the fully qualified path name for the batch file and enclose any arguments to the batch file in double quotes (for example, c:\mydir\myaction.bat "param1 param2").

**Forwarding Traps Between Management Stations**

NetView provides the ability to forward traps from one management station to another. For example, you might have a centralized management station that manages networks for the entire division and management stations at each site for specific locations. The management stations at the individual sites can forward traps to the centralized management station. You can configure the **trapfrwd** daemon to forward specific traps to one or more management stations.

You can also forward traps to the Tivoli IT Director. To use this feature, configure and start the **trapfrwd** daemon. Refer to the *Programmer's Reference* for more information.
Correlating Events

This section describes how event correlation has been implemented on NetView and how it can be used. Correlation is implemented with the nvcord daemon. This daemon, like other NetView daemons, is controlled with an ovsuf file entry which can be modified with nvsetup. The nvcord page in nvsetup is shown as follows:

![Image of NetView Server Setup](image)

Select which of the known rules should be active by highlighting the relevant list box entry. You can also view the contents of a particular ruleset by highlighting the ruleset and selecting View Ruleset. The known rules are the set of *.rs files in the \usr\ov\conf\rulesets directory. File names are standard Windows file names and can contain spaces. Select a file name that effectively describes the ruleset.

For assistance with debugging your customized ruleset, you may select a logging flag from the second list box. Trace normal lists the trap number of each received trap along with passed or failed information for each active ruleset. This should be the default logging level. If this is insufficient, then the additional log flag of rule can be added. This lists the pass or fail information for the individual components of each ruleset.

Implementation

Nvcord registers for trap callback using the standard OVW API and is given a copy of every trap received by the system. It processes these traps according to the specific rulesets active at the time and determines if the rule passes or fails. Everytime a rule is activated, nvcord adds the ruleset name and a rulesetID to a table in the event database (unless the rule is already present), thus providing a database of all ruleset names.

Note: Correlation reduces the amount of raw data (traps) presented to a user; it is assumed that most traps do not pass correlation.
If a rule passes, then nvcord creates a new trap based on the last trap processed by the rule. This trap is given a Source value associated with the active rule. For example, if there was a ruleset consisting of (pass if a node-down event is received for a node that is a member of SmartSet routers) and such a node-down trap arrived, nvcord would create a new node-down trap. The two traps are identical except that the first trap has `Source==Netmon` and the second trap has `Source==RuleSetID`.

Because correlated traps are associated with the triggering rule, you can use the Event Browser to filter by individual rulesets or by all rulesets. The following dialog box shows a new portion of the event browser filter with a request to see all correlated events.

![Figure 20. Portion of Event Browser Filter Dialog Box](image1)

In the next figure, event browser displays the results of the query. Notice that the source column is turned on. This column was displayed previously but had limited usage. It now displays the name of the ruleset that the event satisfied.

![Figure 21. Results of Event Browser Filtering on Source](image2)

It should be noted that filtering by source can be fully combined with any of the other filtering options such as severity and time.

**Rulesets**

The NetView for Windows correlation daemon can use ruleset files created by the NetView for UNIX ruleset editor. Currently, this is the primary means of creating rulesets. The NetView for Windows product includes a number of predefined standard rulesets. However, this does not mean that all functions available in the Tivoli NetView for UNIX ruleset editor are supported in the Tivoli NetView for Windows. Many of the ruleset nodes available in the UNIX version require corresponding code in other parts of the product to carry out their functions and this code is not present in the Windows version. Thus, the following ruleset nodes are not supported in NetView for Windows rulesets:

- Action Node
- Block Node
• Override Status and Severity Node
• Pager Node
• Resolve Node
• Setstate Node

In addition some functions, while supported on Tivoli NetView for Windows, will not work the same way. For example, while In-Line Action nodes are supported on Windows, one cannot use them to launch a Windows console application (one which displays a GUI to the operator), since background processes, such as the correlation daemon, do not have console access in Windows.

The Tivoli NetView for Windows product includes a number of predefined standard ruleset, and these would be used as a guide for the types of operations which rulesets can successfully perform in Tivoli NetView for Windows. To learn more about ruleset editing using the ruleset editor, see the *nvrsEdit*(1) reference page in the *Tivoli NetView Administrator’s Reference* and Creating and Editing a Ruleset in the *Tivoli NetView Administrator’s Guide*.

Two simple ruleset examples are shown below. The first line consists of **Ruleset2 Ruleset** followed by the name of each subrule against which the event should be tested. This example checks to see that a node is a member of the SmartSet routers by using the rule NVQryColl.

```
Ruleset2 Ruleset NVQryColl13
  == 0
  NVQryColl13 NVQryColl
  2\n  "see if node is member of routers smartset"
```

This next rule determines whether the Generic field of the trap is equal to 3 using the EventAttr (event attribute) rule.

```
Ruleset2 RuleSet EventAttr3
  == 0
  EventAttr3 EventAttr
  Generic 0 3 "pass the correlation if generic==3" 0
```

The following example is more complex, passing if a data collector threshold exceeded event is received (trap #58720263) and then no data collector threshold reset event (#58720264) is received within ten minutes. Notice on the first line how there are two rules mentioned (TrapID3, TrapID4). This indicates that all traps should be tested against both rules. Each rule in turn checks the enterprise ID and specific trap number of each trap. Notice also the second to the last line which contains AttrJoin. This means that these two rules must be combined for the whole rule to pass.

```
RuleSet2 RuleSet TrapID3 TrapID4
  == 0
  TrapID3 TrapID4 AttrJoin5
  metview 1.2.3.4.1.2.3.8 "5" "SN24263" 0 "metview" "CallData|Threshold Specific SN24263": 0
  TrapID4 TrapID4 AttrJoin5
  metview 1.2.3.4.1.2.3.8 "5" "SN24264" 0 "metview" "CallRearm|Threshold Specific SN24264": 0
  AttrJoin 0 "" 0 "2": 0
```

*Figure 22. Ruleset*

### Paging Support

NetView paging support consists of two parts: a program to submit a pager request and a daemon that processes pager requests by dialing a paging carrier and transmitting the page.
The `nvpage` command is used to submit a page to the system–wide pager request queue. The system–wide pager request queue is maintained by the nvpagerd daemon.

The nvpage daemon (Nvpagerd) is run on the system that has telephony hardware installed and the Windows telephony settings configured to use the telephony hardware.

You can configure an event to issue a call to a pager when the event is received. To do this, select `options → trap settings` and enter the `nvpage` command as an executable command in the trap settings dialog.

The `nvpage` command can be executed on a system in which the pager request daemon is not running as long as the name of the system on which the pager request daemon is running is supplied with the `-h` option.

For example, to send a page to an alphanumeric pager 1116676 receiving pages through the carrier Caltel in an environment where the data modem is installed on a system named fred.mynet.com, use the command:

```
nvpage -h fred.mynet.com 1116676@caltel Meet at 8:00am in the lobby
```

The following command sends the numeric message 4085551234 to the numeric pager that is identified by the number 5558899 through dialing Caltel’s 800 number:

```
nvpage-n 5558899@caltel 4085551234
```

The following command sends the numeric message 4085551234 to the numeric pager that can be dialed directly with the number 5556789:

```
nvpage 5556789 4085551234
```

**Configuring Your System to Use the nvpage Command**

To use the `nvpage` command, install telephony hardware that provides `comm/datamodem` capabilities.

1. Following the telephony hardware manufacturer’s instructions for adding the equipment to your system.

2. 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 Model 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.

   - `nvpaging.protocols`
     - Defines the characteristics of the following paging protocols: TAP, PAKNET, DIRECT_INPUT, IXO, PET. You should not need to modify this file.

3. Invoke the NetView Server Setup application from the Start menu *(Start—> Programs—> NetView —> Administration —> Server Setup)*.

4. From the Daemons page, use the drop-down list and select the pager daemon.
5. On the pager daemon configuration page, select the name of a paging carrier that will be used and then specify the modem that will be used to call that paging carrier. Click the Configure... button that is to the right of the drop-down list box containing the selected modem. If the "default modem" is selected, then the modem configuration specified as the default modem in the system control panel will be used to contact the paging carrier.

6. In the modem configuration dialog box, specify the modem settings such as baud rate, data bits, parity and stop bits recommended by your paging carrier. If you are unsure what values to supply for these settings, contact your paging carrier.

7. Click OK to confirm the modem configuration dialog box and click OK again to confirm the nvpagerd configuration.

Accepting Phone Numbers and Using TAPI Settings

If the nvpage command is given a fully-qualified phone number, such as: 1-603-555-1212 the command applies the settings for the dialing properties of the current modem to dial the phone number. This means that dialing prefixes are dialed as described in the dialing properties dialog box of the current modem configuration.

To avoid using the dialing properties of the current modem configuration, enter the number as you would like to have it dialed. For example, if you are dialing from within the 603 area code, but your telephone system requires that you dial a special prefix, such as the number 8, for phone numbers outside your local calling area but still within the 603 area code, you can override the system telephony settings by using the phone number 816035551212 as an argument to the nvpage command.

Additionally, some paging services provide a voice menu system, which must be used to send a page. For example, if the paging carrier to be dialed had a voice announcement that said, "press 1 followed by # to send a numeric page or press 3 followed by # to send a voice page," and you desired to send a numeric page through this carrier, you can use a dial string such as:

nvpage 5551212 1#,5551611 the leading "1#" characters make the correct choice in the paging carrier’s telephone menu system to navigate to the numeric paging service. The ",," characters tell the telephony hardware that is dialing the paging carrier to wait 5 seconds for each ",," character or ten seconds all together before dialing additional digits.

By default, the ",," character is used to instruct the modem dialing a numeric pager to pause for five seconds before proceeding with the dialing operation. Although the ",," is recognized by a wide variety of modems, it is not recognized by all modems. If you find that your modem does not recognize the ",," as the dial string wait character, then you can override the default and specify the character that is recognized by your modem through the registry variable PagerWaitString. Create this variable through

HKEY_LOCAL_MACHINE\SOFTWARE\Tivoli\NetView\CurrentVersion

Pager Configuration Files

Files which store paging carrier modem configurations are stored in \usr\ov\conf. The base name of the file corresponds to the name of the paging carrier. The file extension is ".conf".
Dealing with Errors

If you are experiencing difficulty connecting to a paging carrier, ensure that you have properly configured the modem properties to reflect the settings that the paging carrier is expecting.

Note that failed pages are dumped from the system paging queue after an excessive number of retries. For example, if the paging carrier phone number is busy for an extended period of time, the page will be dumped if a connection cannot be made to the paging carrier after seven attempts.

Modems configured with initialization strings that are not valid will also cause pages to fail and be dumped. Dumped pages are placed in the file \usr\ov\log\nvpager.dumped.
Chapter 7. Managing the Network

The NetView program 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 in any of the following ways:

- Monitoring the network for signs of potential problems
- Resolving network problems
- Collecting information for trend analysis
- Allocating network resources
- Planning future resource acquisition

The performance data collected by the NetView program is based on the values of MIB objects. The NetView program provides applications that display performance information:

- NetView Graph displays MIB object values in graphs.
- Other NetView tools display MIB object values in tables or forms.

This chapter describes MIBs and explains how to load, unload, and browse them. It also describes how to use the NetView program’s predefined performance applications and how to create your own applications to monitor network performance. The following topics are covered in this chapter:

- "Understanding Network Management Protocols" on page 74
- "Understanding MIBs" on page 75
- "Loading and Unloading MIBs" on page 75
- "Browsing MIBs" on page 76
- "Using the NetView Performance Applications" on page 77
- "Monitoring Real-Time Network Performance" on page 78
- "Collecting Historical Performance Information" on page 81
- "Using the NetView Graph Applications" on page 81
- "Using Other Tools" on page 84

Understanding Network Management Protocols

To manage a network, you need the following:

- A network management system, such as the NetView program, that executes network management applications that collect, process, store, and display network data.
- A system that contains agents (programs) that respond to requests made by the management system.
- Information about the physical and logical characteristics of the network objects that are part of the managed system.
- A method for the network management system and agent to share and process information about the network objects.

The NetView program manages TCP/IP networks by using the Simple Network Management Protocol (SNMP). SNMP enables management systems to ask agents to retrieve and change information about network objects.

SNMP provides commands that management systems can use to communicate with agents about the status of network objects. The SNMP commands are Get and Set.
• Get: requests that the responding agent supply the values of one or more of the variables of the object managed by the agent. For example, management systems can request a description of a particular network object.

• Set: writes new data to one or more of the variables of the object managed by an agent. For example, the management system might need to update information about the network object whose description was provided to you by a Get command.

Note: The Windows SNMP agent does not support Set commands.

Some of the NetView applications and menu items run these commands and display the results using NetView windows. You do not need to use these commands directly.

Understanding MIBs

The physical and logical characteristics of network objects make up a collection of information called the MIB. The MIB is not an actual database residing somewhere on the network. The individual pieces of information, called MIB objects, reside on the agent system, where they can be accessed and changed by the agent at the manager’s request. This is how the NetView program manages network objects.

MIB objects and network objects are not the same. Network objects are particular devices that reside on a network. MIB objects are individual pieces of information that, taken as a whole, describe a network object.

MIBs Supported by the NetView Program

The NetView program supports the following kinds of MIBs:

• Standard MIB: all devices that support SNMP are also required to support a standard set of common managed object definitions, of which a MIB is composed. The standard MIB object definitions, MIB-I and MIB-II, enable you to monitor and control SNMP managed devices. Agents contain the “intelligence” required to access these MIB values.

• Enterprise-specific MIB: SNMP permits vendors to define MIB extensions, or enterprise-specific MIBs, specifically for controlling their products. These enterprise-specific MIBs must follow certain definition standards, just as other MIBs must, to ensure that the information they contain can be accessed and modified by agents.

The NetView program provides the ability to load enterprise-specific MIBs from a MIB description file. By loading a MIB description file containing enterprise-specific MIBs on a manager station, you can monitor and control vendor devices.

• Generic topology MIB : the NetView program provides a generic MIB definition that enables you to manage network elements that communicate by using a protocol other than SNMP. For a list of protocols defined by the NetView program, refer to Tivoli NetView for Windows Programmer’s Guide.

MIB Naming Conventions

MIB objects are logically organized in a hierarchy called a tree structure. Each MIB object has a name derived from its location in the tree structure. This name, called an object ID, is created by tracing the path from the top of the tree structure, or the root, to the bottom, the object itself. Each place where the path branches is called a node. A node can have both a parent and children. If a node has no children, it is
called a leaf node. A leaf node is the actual MIB object. Only leaf nodes return MIB values from agents. Figure 23 shows a MIB tree structure.

A full MIB object ID contains all nodes, starting at the root, and including the leaf node. The nodes are concatenated and separated by periods, in a format known as dotted decimal notation. For example, the mib-2 subtree is .iso.org.dod.internet.mgmt.mib-2, which is concisely written as .1.3.6.1.2.1.

The standard MIB-I and MIB-II definitions are registered in the mib-2(1) subtree. The mib-2(1) subtree is primarily used to manage TCP/IP-based networks through the SNMP protocol. See the Tivoli NetView for Windows Programmer's Guide for more information on the mib-2 subtree.

Enterprise-specific MIBs are registered off the private(4) branch in the subtree enterprises(1). Each enterprise is assigned a number. Digital is assigned the enterprise number 36; so all Digital enterprise-specific objects have the dotted decimal notation starting with .1.3.6.1.4.1.36, corresponding to the tree structure .iso.org.dod.internet.private.enterprises.digital. IBM is assigned the enterprise number 2, so all IBM enterprise-specific objects have the dotted decimal notation starting with .1.3.6.1.4.1.2, corresponding to the tree structure .iso.org.dod.internet.private.enterprises.IBM.

### Loading and Unloading MIBs

The Tools..MIB..Load command lets you include your enterprise-specific MIB in the NetView program's loaded database.
The purpose of loading a MIB is to define the MIB objects so the NetView program's applications can use those MIB definitions. The MIB you are interested in must be loaded on the system where you want to use the MIB Browser, MIB Data Collector, MIB Tool Builder, and the applications built by the MIB Tool Builder. 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.

To load an enterprise-specific MIB, copy the MIB into the default directory, \usr\ov\snmp_mibs, then select Tools..MIB..Load. If you know the full path name of the directory where the MIB is located, you can select Tools..MIB..Load and enter the full path name in the 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.

After 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

Select Tools..MIB..Load to unload MIBs from the MIB database.

NetView also supports SNMP V2 MIBs. To load or unload SNMP V2 MIBs, select Tools...MIB...SNMP V2...Load.

---

**Browsing MIBs**

Use the MIB Browser to query MIB values for both 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*.

The MIB object iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifDescr, for example, has as many instances as its associated network object has interfaces, as shown in the following example:

1: SE0 DESVA-Class Ethernet Device.
2: L00 Loopback Port.

However, the system MIB object iso.org.dod.internet.mgmt.mib-2.system.sysContact has only one instance per network object, because generally there is only one person designated as the system contact. This is MIB instance 0, as shown in the following example:

sysContact.0: J. J. Yukio 555-1234

On a submap, select a network object whose MIB objects you want to view. Select Tools..MIB..Browser from the NetView main menu.

Select Tools..MIB..Browser to save MIB data in a file for use in reports or by another application.
NetView also supports SNMP V2 MIBs. To view SNMP V2 MIBs, select Tools...MIB...SNMP V2...Browser.

Using the NetView Performance Applications

The NetView program provides applications that enable you to monitor both real-time and historical network performance. Some predefined applications are described in "Using NetView Predefined Applications" on page 78.

In addition, the MIB Tool Builder enables you to create applications that 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. Table 13 compares the MIB Tool Builder and the MIB Data Collector.

Table 13. Comparison of NetView MIB Applications

<table>
<thead>
<tr>
<th>Task</th>
<th>MIB Tool Builder</th>
<th>MIB Data Collector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting</td>
<td>Select Tools...MIB...Tool Builder</td>
<td>Select Tools...MIB...Collect Data</td>
</tr>
<tr>
<td>Selecting MIB</td>
<td>Can select more than one MIB variable</td>
<td>Can collect data for only a single MIB instance</td>
</tr>
<tr>
<td>Variables</td>
<td>Can apply selection rules to MIB variables</td>
<td>Can specify a single interface, but data is collected on the MIB instance</td>
</tr>
<tr>
<td>Specifying Polling</td>
<td>Can specify polling intervals only for Graph format</td>
<td>Can change polling intervals for individual nodes</td>
</tr>
<tr>
<td>Intervals</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Setting Thresholds</td>
<td>Not applicable</td>
<td>Can choose one of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Exclude Collection</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Store, Check Thresholds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Store, No Thresholds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Don’t Store, Check Thresholds</td>
</tr>
<tr>
<td>Generating Events</td>
<td>Not applicable</td>
<td>Can generate events and define specific trap numbers</td>
</tr>
<tr>
<td>Selecting Output</td>
<td>Choose table, form, or graph</td>
<td>Shows data in table format. You can select Graph from the table display.</td>
</tr>
<tr>
<td>Format</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customizing Output</td>
<td>Can customize the “name” of the MIB variable displayed in the title bar</td>
<td>Uses applications in the \usr\ov\reports\ic\ directory to graph output and generate reports</td>
</tr>
<tr>
<td></td>
<td>Cannot customize line names</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Customization of graphic display lost between invocations</td>
<td></td>
</tr>
<tr>
<td>Saving Output</td>
<td>Can save output to file. Default is \usr\tmp\monitor.txt</td>
<td>Collection file always stored in \usr\ov\databases\snmpCollect</td>
</tr>
<tr>
<td>Exporting to</td>
<td>Cannot export graph data to ASCII format</td>
<td>Can convert to ASCII format</td>
</tr>
<tr>
<td>ASCII Format</td>
<td>Can export summary instance table format</td>
<td>Time points provided through NetView Setup</td>
</tr>
<tr>
<td></td>
<td>No time points are provided</td>
<td></td>
</tr>
</tbody>
</table>
Table 13. Comparison of NetView MIB Applications (continued)

<table>
<thead>
<tr>
<th>Task</th>
<th>MIB Tool Builder</th>
<th>MIB Data Collector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performing Other Calculations on MIB Variables</td>
<td>Not applicable</td>
<td>Apply expressions in $usr$ov$conf$mibExpr.conf file to MIB variables</td>
</tr>
</tbody>
</table>

For more information about the MIB Tool Builder, see [Working with MIB Applications](#) on page 80. For more information about the Data Collector, see [Collecting Historical Performance Information](#) on page 81.

### Monitoring Real-Time Network Performance

The NetView program provides applications that enable you to monitor real-time network performance. It also provides you with the MIB Tool Builder, which enables you to build your own applications. This section describes the NetView program’s predefined applications and the MIB Tool Builder.

#### Using NetView Predefined Applications

The NetView performance applications collect data and display it in table, form, or graph format. See the following sections for individual application descriptions and online help for information on how to use the applications.

**Monitoring Interface Traffic**

If you suspect network performance problems, select `Monitor..Graph..Interface Traffic` from the NetView main 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 Ethernet graphing options listed under `Monitor..Graph` on the NetView main menu.

**Monitoring Ethernet Traffic**

Select `Monitor..Graph..Ethernet Traffic` from the NetView main menu to monitor and graph interface statistics for an HP node that supports SNMP.

The following information is displayed in the graph:

- The number of packets transmitted successfully by the interface hardware
- The number of packets received successfully by the interface hardware
- The number of bits per second transmitted successfully
- The number of bits per second received successfully

**Monitoring Ethernet Errors**

To view interface error statistics for a gateway server that supports SNMP, select `Monitor..Graph..Ethernet Errors` from the NetView main menu. This command graphs the average number of packets per second as computed over the previous polling interval for the following statistics:

- The number of packets received with a cyclic redundancy check (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**
Select *Monitor..Network Properties..TCP Connections* from the NetView main menu to list the TCP connection table for selected remote SNMP nodes. This table shows the local and remote addresses for each node and the state of each session.

**Monitoring SNMP Networks**
You can monitor and graph the performance of SNMP networks by using the commands listed under Monitor on the NetView main menu.

**Monitoring SNMP Traffic**
Select *Monitor..Graph..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**
Use the *Monitor..Graph..SNMP Operations* menu item to display a summary of the SNMP operations requested of and performed by the SNMP agent on the selected nodes. The operations reported are the total 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 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**
Select *Monitor..Graph..SNMP Errors* to display a summary of the SNMP protocol errors detected by the SNMP agent on the selected nodes. The errors reported are 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.

You can obtain a list of the management systems that have caused authentication failures on a selected node that is an HP node. The SNMP authentication feature queries an enterprise-specific MIB.

Select Monitor..Network..SNMP Authentication Failures from the NetView main menu. 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.

Working with MIB Applications
The NetView program enables you to build, modify, and delete MIB applications using the MIB Tool Builder.
• Build MIB Applications: you can build MIB applications without programming when you want to monitor the real-time performance of specific MIB objects or poll certain MIB objects on a regular basis and produce output such as forms, tables, or graphs. To build a MIB application, select Tools..MIB..Tool Builder..New.
  By default, the new MIB application is placed under Monitor..MIB Properties in the NetView main menu. To look at MIB values returned by the new application, select the appropriate item under the Monitor option on the NetView main menu.
• Modify MIB Applications: using the same Tools Builder dialog box, you can modify an existing MIB application. Select the application, then click on the Properties button.
• Delete MIB Applications: to delete a MIB application from the Tool Builder dialog box, select the MIB application and click on the Delete button.

Running MIB Applications within a Map
To run your application, select an object or objects on the network map, select the menu option you associated with your application from the NetView main menu, and follow the menu path you entered for the application. Your application will be started in a separate window and will run until you stop it by selecting Close.

If a MIB application does not run, check the following conditions:
• Ensure that the MIB is loaded into the Loaded MIBs 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.
• Verify 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\ovsnmp.conf` file to verify the community name.

## Collecting Historical Performance Information

You can compile historical performance data about your network by using the 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.
• 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.

Before configuring your system to collect MIB data, you should understand the definitions of the MIB objects and what they do. To see descriptions for selected MIB objects, select **Tools...MIB...Browser** or **Tools...MIB...SNMP...V2 Browser** from the NetView main menu. Refer to the vendor documentation for information about a vendor’s enterprise-specific MIB.

To access the NetView data collection features, select **Tools..MIB..Collect Data** from the NetView main menu. If no data is being collected, ensure that the snmpCollect daemon is running. This daemon stops running when file system space is not available.

Ensure that there is enough room to store data in the SQL database. You can periodically purge entries by selecting **Options..Server Setup**, the **Files** tab page, then **Schedule SNMP Files to Delete** from the drop-down list.

After the problem is solved, select **Options..Server Setup** to restart the daemon. Then, you can restart the data collection. Refer to the *Tivoli NetView for Windows Programmer’s Reference* for more information about the snmpCollect daemon and the `snmpodump` command.

## Using the NetView Graph Applications

The NetView Graph tool displays real-time traffic data, real-time SNMP data, and collected data in a graph format. NetView Graph provides you with a convenient way to display numerical performance information.

## Starting a Graph Application

To start a graph application, select an object or objects on a submap, then select the graph application that provides results you want to view. All graph applications can be started from the NetView main menu. The graphs are displayed in separate windows.

The following graphs can be started from the NetView main menu:
• Monitor..Graph..Interface Traffic
• Monitor..Graph..SNMP Traffic
• Monitor..Graph..SNMP Operations
The following graphs can be started from the NetView Application toolbar:

- Interface Traffic
- Collected Data

You can also start the NetView Graph from within dialog boxes. The NetView Graph tool graphs the data supplied by the dialog box. The dialog boxes, where you can start the NetView Graph tool, are:

- MIB Browser
- MIB Browser SNMP V2
- Collections..Data..MIB Data

### Printing Graphs

To print a graph, follow these steps:

1. Select **Edit..Copy** from the NetView Graph menu.
2. Run the Paintbrush program and paste the graph into a Paintbrush file.
3. Select Print from the Paintbrush File menu.

### Customizing the NetView Graph Tool

Graph applications have certain default display 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 24** shows the NetView Graph tool’s standard layout for graph applications, using the **Monitor..Graph..Interface Traffic** performance application.
Setting Time Intervals
The NetView Graph tool displays the statistics associated with the graph in the legend at the bottom of the NetView Graph window. The x-axis shows the time the data is collected, and the y-axis identifies the data. For example, the y-axis title is Packets for the Interface Traffic graph.

By default, the NetView Graph tool displays 10 minutes of results. You can change the amount of time shown on the graph by selecting Graph Properties from the NetView Graph Edit menu, and then changing the Time Visible Range. After making your changes, click on **OK** or **Apply**. Notice that the graph display and the end time change.

The Poll Nodes Every field in the Graph Properties dialog box enables you to set how often a node is polled for information and how often information is displayed in the graph. Poll Nodes Every has no impact on the display of collected data.

Changing Line Properties
You can change the properties of the lines on the graph. Select a line in the legend below the graph, and then select **Edit...Line Properties** from the NetView Graph menu. In the Line Properties dialog box (Figure 25), you can change the width and color of the line and its multiplier.

![Figure 25. Line Properties Dialog Box](image)

The Line Properties dialog box displays a row of fields for a particular line on the graph. The fields of the Line Properties dialog box are described in the following list.

- **Width** You can change the line width by selecting Small, Medium, or Large from the drop-down list.

- **Color** You can change the color of each line by selecting a color from the drop-down list.

- **Multiply MIB Values by**
  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 without multipliers.
Scaling Data along the Y-Axis
You can scale all data or just visible data along the y-axis. The default is to scale all data. Scaling all data displays the greater fluctuations that may occur from high to low values. If you need to base decisions on the overall pattern of data, continue to scale all data. If you want to smooth out the display of data, change the scale to display visible data.

To change the scale:
1. Select Edit..Graph Properties from the NetView Graph menu.
2. In the Graph Properties dialog box, select the General tab page.
3. In the General tab, select Visible Data from the MIB Value Range Matches drop-down list.
4. Select OK.

Displaying or Hiding the Grid
By default, the NetView Graph tool displays graphs without grids. You can choose to display a grid by selecting View..Grid from the NetView Graph menu.

Using Other Tools
The Tools..Other command allows you to run applications within the NetView program. For example, you can create applications that collect and display the performance information you need to see, and then run those applications by selecting Tools..Other from the NetView main menu.

When you want to run a report or an application:
1. Select one or more nodes from a submap.
2. Select Tools..Other from the NetView main menu.
3. Select a file name.
4. Select OK.

Tools..Other passes information about the selected nodes to the application.

Tracking Availability of Nodes
NetView keeps track of when nodes go up and down and records these transactions in the ODBC database. You can view this data, along with statistics, such as MeanTimeBetweenFailures, in the following ways:
• The Submap Explorer Availability View enables sorting and filtering availability data based on time.
• The Availability page in the Object Properties dialog. This page shows availability information for the selected node and enables time-based filtering of this information. This data can be saved in a text file or an HTML file.
• The uptimer.exe command line utility. Use this utility to dump the availability tracking data.
• The Visual Basic program sample, Availability, located in \usr\ov\prg_samples\availability. This program sample contains both the source code for the application and a separate setup program to install all files needed to run the compiled Visual Basic application.

Note: This program sample works only on Intel platforms and Microsoft Access databases. The program sample has not been tested with Alpha or with Microsoft SQL Server.
Chapter 8. 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:

- Ensure 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.

The following topics are covered in this chapter:

- "Discovering the Network"
- "Monitoring Network Configuration" on page 90
- "Retrieving MIB Configuration Information" on page 92
- "Setting and Changing Polling Intervals" on page 93
- "Configuring SNMP Nodes" on page 94
- "Configuring SNMP Timeout Intervals" on page 95
- "Discovering Services on Nodes" on page 95

Discovering the Network

Network discovery provides you with a database of network configuration information. An IP network can be discovered using automatic network discovery or by using a seed file to control network discovery.

Automatic Network Discovery

The NetView program 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.

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 shows the information that is retrieved.

Table 14. Network Discovery Configuration Information

<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 network hierarchy.</td>
</tr>
</tbody>
</table>
Table 14. Network Discovery Configuration Information (continued)

<table>
<thead>
<tr>
<th>Information</th>
<th>MIB Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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 contact person for this network object and tells how to contact that person.</td>
</tr>
</tbody>
</table>

After the node has been discovered, these MIB values are polled periodically. Any changes are reflected in the topology database. You can display this information for a node by using the MIB Browser.

You can store additional, enterprise-specific information with each node and network. This information, used with the predefined generic data generated by the NetView program, can give you a clearer picture of your network’s configuration. To store or load additional MIB information, use the MIB Loader.

**Using Alternative Community Names**

It is often the case that different regions, classes of devices (such as routers), or management domains use different community names. You can add a list of up to six SNMP community names, which are used by various devices in your network, to the file `\usr\ov\conf\communityNames.conf`. When a netmon query to a device fails, netmon will try each of the community names in this file, in an attempt to find the community name that works. An entry is then created for that node name/community name combination in the ovsnmp.conf file and the entry will be used in future queries.

Normally, netmon will use the community name from the SNMP configuration. If a community name has not been configured for a device, then it will use public, the default. If this fails, netmon will use the list in communityNames.conf, but only under the following circumstances:

- On initial discovery for all nodes.
- After SNMP timeouts for routers netmon will always try names from the list.
- When a demand poll fails, netmon will always use the list.
- A netmon configuration option allows netmon to use the alternate community name list and update its internal cache during its configuration polling, if necessary. This option should be used judiciously because it will slow down polling noticeably due to timeouts for non-SNMP nodes. Ideally, you would want to use this option when a large number of community strings have been changed. It is intended to be used during a full configuration cycle (which is usually 24 hours), and then turned off again.
To enable this configuration option, select the **Discovery** page in the Server Setup application. Select **Always** in the combo box in front of the **Use Alternative Community Names in File** field.

**Discovering Cisco Routers**

NetView 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.

**Off Loading Discovery to Remote MLMs**

NetView for Windows can now act as a Regional Manager to offload discovery and status polling (ping) to specific subnets, or segments within subnets by working with Mid-Level Managers (MLMs) installed on the network. The following sections describe how NetView for Windows works with MLMs.

It is recommended that you not run the MLM on the same node as NetView to avoid conflicts with their respective event subsystems.

**Polling**

The netmon daemon will not discover nodes using SNMP or ping sprays nor status poll nodes in a region that is specified to be managed by an MLM. Instead, the MLM performs discovery and status polling and conveys the information to NetView using the following MLM enterprise traps: MLM_NewNodes, MLM_StatusDown, and MLM_StatusUp.

The netmon daemon periodically polls the MLM nodes for their Discovery table as defined in the Mid-Level Manager for Windows MIB Definition, which is located in the `<usr\ov\snmp_mibs\mlm.mib>` file. No SNMP requests for ARP Table information or Routing Table information are made to the MLM node unless the MLM node has interfaces to other networks. The netmon daemon stills polls nodes in regions that are managed by MLM nodes for configuration data using SNMP to accurately represent the MLM-managed nodes on the map.

**Subnet Masks**

The subnet mask of the MLM node is needed to accurately define the region being managed by the MLM node. An assumption made by the netmon daemon is that all nodes within a subnet being managed by an MLM node will have the same subnet
mask as the MLM node itself. Hints about nodes in a subnet managed by an MLM node that do not have a matching subnet mask will be ignored. Note that nodes with inconsistent subnet masks are logged in the \texttt{usr\log\badmasks.log} file (if enabled), and that Inconsistent Subnet Mask events are logged in the Event Browser. If the node is successfully added to the topology (for example, its subnet mask is at least well formed), then the BadMasks SmartSet should contain such nodes as well.

**Map Operations**

It is recommended that regions being managed by MLM nodes \textit{not} be unmanaged then managed, as the map status of such nodes may remain UNKNOWN for extended periods of time, leading to confusion. This is because netmon does not status poll the MLM-managed nodes and the MLM only sends status events when status changes. To force netmon to update nodes, select the \texttt{Test --> Demand Poll} or \texttt{Test --> Ping} operations. The netmon daemon ignores the manage and unmanage actions on nodes in MLM-managed regions.

The \texttt{Object --> Discover} option has no affect on subnets being managed by an MLM node.

**SmartSets**

When netmon is instructed to recognize nodes running MLMs, it will set the \texttt{"isMLM"} field for those nodes in the ovw object database. Such nodes will then be displayed in the MLM’s SmartSet.

**How to Enable NetView as Regional Manager**

See the Mid-Level Manager for UNIX or Windows documentation on how to setup the MLM to forward traps to the Regional Manager running NetView for Windows.

To activate netmon as a Regional Manager:
1. Select \texttt{Tools --> MIB --> Load} and load the mlm.mib file.
2. Select \texttt{Options --> Server Setup --> Discovery} to enable MLM support.

   The netmon daemon can learn about the nodes running MLM from an MLM nodes seed file or by recognizing traps received from MLMs or both.

   If you select the MLM nodes seed file method, edit the MLM seed file and enter the IP addresses of the nodes running the MLM you are interested in.

After netmon is instructed to recognize MLM nodes, netmon assumes the role of regional manager. If netmon is not instructed to recognize MLM nodes, then netmon discontinues its role as a regional manager, even if MLM nodes exist in the topology. When netmon is not acting as a Regional Manager, nodes that had been managed by an MLM node will now be actively managed by netmon, and netmon resumes discovery and status polling.

**NetView as an Attended MLM**

NetView can function as a mid-level manager monitoring and reporting on the network in the same manner as the MLM for Windows product. This reduces duplicate network polling for regional managers who are also managing the network from a local site. It also allows the Central Manager to view the same map and SNMP data at the same time as the Regional Manager.

The Central Manager configures and reads the MLM MIB in the same manner as any other MLM for or UNIX. Currently, the central manager would use the NetView for UNIX smconfig tool or the APM component of NetView for UNIX. The MLM provides basically four functions for the central manager at the local subnet level.
1. Discovery
As NetView discovers new nodes in the local subnet it reports them in the MLM Discovery table and sends off the MLM NewNodes trap to the addresses specified in the MLM Trap Destination Table.

2. Status Monitoring
NetView maintains the status of the nodes in the MLM Interface Status Table according to the configuration in the MLM Status Monitor Table. When a change of status is detected, NetView sends an MLM_StatusUp or MLM_StatusDown trap to the addresses specified in the MLM Trap Destination Table.

3. SNMP Data Collection and Thresholding
Thresholding and data collection policies are configured in the MLM Threshold and Collection table. When a threshold or rearm condition is matched, the MLM ThresholdArm or MLM ThresholdReArm trap is sent to the addresses specified in the MLM Trap Destination Table. If data is to be stored, it will be stored in NetView's SQL databases along with other SNMP data collections configured using NetView itself. A log entry will be made for each collection in the MLM Data Collect Table.

4. Trap Filtering and Forwarding
NetView can forward all traps to the addresses specified in the MLM Trap Destination Table that match the filters specified in the MLM Filter Table.

For information on how to enable NetView as an Attended MLM, refer to the \usr\ov\doc\netview_mlm_readme.htm document.

Using a Seed File to Control Network Discovery
When the NetView program is started for the first time, the default management region is the management system on which the NetView program is operating and any networks to which it is attached. The discovery process generates the topology map by working outward from the management system.

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 the discovery process to generate the topology map beginning from nodes other than the management system.

The format of the seed file is a list of host names and IP addresses of SNMP nodes within your administrative domain. Each host name or IP address must be on a separate line. Comments, preceded by a pound sign (#) are permitted after the host name or IP address on a line. Entries that begin with an exclamation point (!) define an IP address or range of IP addresses to be excluded from discovery (a negative entry). The $ prefix specifies nodes for which SNMP should be used to query for status instead of ICMP pings. The % prefix indicates HSRP interfaces. The following example shows the format of a seed file:

core1.division.company.com
ROUTer4.division.company.com #Gateways make the best seeds.
9.67.1.5
9.67.5.119
!9.63.*.* #Exclude this range of addresses from discovery.
!9.67.5.120 #Exclude this IP address from discovery.
@1.1.1.1-9 # DHCP addresses
$1.1.2-10.221-230 # SNMP Status entries
%1.1.1.1 # HSRP Interface
The seed file also supports the management of Dynamic Host Configuration Protocol (DHCP) nodes in your network entries which begin with the @ character signifying DHCP addresses. See the seed file template supplied by the NetView installation for the rules regarding entering DHCP ranges. The template is \usr\ov\conf\netmon.seed.

To edit the seed file, select Options..Discovery from the main menu. From here, you can also limit discovery to nodes in the seed file. Check the Discover with Seed File option and select Edit to launch the Network Monitor Seed File Editor. You can use this editor to add, delete, or edit seed file entries.

### Using Network Aliases

You can assign names and aliases to specific networks and subnetworks using the %systemroot%\system32\drivers\etc\networks file. Place an entry in the file for each network or subnetwork you want to assign an alias. Each entry in the file represents one network. When the NetView program discovers the network, it displays the network on the map using the name from this file as the label.

The format of the file follows:

```
<network name> <network number> [aliases...] [#<comment>]
```

For example:

```
HeadQtrs  284.122.107
Campus    284.122.108
```

### Monitoring Network Configuration

There are many reasons to keep track of the configuration of objects on your network. The following 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.

### Displaying Addresses for Remote SNMP Nodes

To display IP and non-IP addresses associated with a remote SNMP node, select a node on a submap, then select Monitor..Network..Addresses from the NetView main menu to collect 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 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.
Displaying Configured Interfaces

Select Monitor..Network..Interfaces from the main menu to display 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.

The Monitor..Network..Interfaces operation provides the following information:

- The 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. If the status is 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 MIB table corresponds to a port on the hub or bridge.

Displaying Routing Table Information

To display routing table information for selected remote SNMP nodes, select Monitor..Network..Routing Table from the main menu. The following information is provided to help you resolve connectivity problems.

- 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 following types of connection:
  - Direct, 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.

Displaying 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 display this information for selected remote SNMP nodes, select Monitor..Network..ARP Cache from the NetView main 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
Displaying Configured IP Services

You might need to know the IP services a node is configured to support. For example, a user might be having trouble accessing a particular service on a remote SNMP node. Select Monitor..Network..IP Services to learn the following information:

- The service protocol.
- The port to which the service is bound.
- The service for which the node is listening, such as SNMP or Telnet. If this field is blank, the service is unknown.

Retrieving MIB Configuration Information

Certain MIB variables store information that provides a summary of the configuration of a selected network node or nodes. The NetView commands 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 select Monitor..System..Interface from the NetView main menu. See Table 15 for additional information.

Table 15. MIB Interface Information

<table>
<thead>
<tr>
<th>MIB Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ifDescr</td>
<td>The manufacturer, product, and version of the hardware interface</td>
</tr>
<tr>
<td>ifType</td>
<td>Type of interface</td>
</tr>
<tr>
<td>ifMtu</td>
<td>Size of largest datagram that can be sent or received over this interface, specified in octets</td>
</tr>
<tr>
<td>ifSpeed</td>
<td>Estimated current bandwidth of the interface, in bits per second (bps)</td>
</tr>
<tr>
<td>ifPhysAddress</td>
<td>Address of the protocol layer at the protocol layer immediately below the network layer in the protocol stack, in octets.</td>
</tr>
<tr>
<td>ifAdminStatus</td>
<td>Desired state of the interface (up, down, or testing)</td>
</tr>
<tr>
<td>ifOperStatus</td>
<td>Current operational state of the interface</td>
</tr>
</tbody>
</table>

Displaying MIB System Information

To display the current values of a network object’s system MIB variables, select a symbol on the submap, then select Monitor..System..General from the NetView main menu. The information in Table 16 is displayed:

Table 16. MIB System Information

<table>
<thead>
<tr>
<th>MIB Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysDescr</td>
<td>Name and version of the hardware, software operating system, and networking software</td>
</tr>
<tr>
<td>sysObjectID</td>
<td>Node that is being managed</td>
</tr>
<tr>
<td>sysUpTime</td>
<td>Time, in hundredths of a second, since the network management portion of a system was last initialized</td>
</tr>
<tr>
<td>sysContact</td>
<td>Contact person for this managed node, and how to contact that person</td>
</tr>
</tbody>
</table>
### Monitoring the Network Using Polling

NetView has comprehensive polling capabilities that enable you to view the up-to-date status of your network. NetView can use PING (ICMP echo) requests to poll devices for status, or use SNMP GET requests to determine the status of all interfaces on a device.

Many aspects of the NetView polling configuration can be customized to fit your network environment. The following sections describe how to customize polling for your network.

### Setting and Changing Polling Intervals

To turn polling on or off and to set polling intervals for all IP nodes in the management region, select **Options..Polling**. When you use **Options..Polling**, the polling intervals apply to all nodes.

You cannot use this operation to set polling intervals for individual nodes. If you want to set or change polling intervals for individual nodes, select either **Tools..MIB..Collect Data** or **Options..SNMP** from the NetView main menu. See "[Collecting Historical Performance Information](#)" on page 81 for a description of the **Tools..MIB..Collect Data** option. See "[Configuring SNMP Nodes](#)" on page 94 for details about **Options..SNMP**.

To check the status of the NetView processes that do the polling, select the **Daemons** tab page of the Polling Options dialog box. In the Status page, you can specify how often the NetView processes are checked, and stop or start automatic checking for any of the processes.

### Scheduling Off-Period Polling

You can now enable weekly off periods for status, configuration, and discovery polling. For example, you might not care about polling over the weekend for certain subnets or ranges of nodes. To schedule off periods for polling, select **Options --&gt; Polling**. On the Polling Options dialog, enable Polling Off Periods and click on the **Edit** button to edit the default file, `usrlovconf/offperiods.conf`. The format of this file is as follows:

```
<IPAddress> <StartDay> <StartTime> <EndDay> <EndTime> <PollTypes>
```

Where:

- **IPAddress**
  Specifies the IP address, expressed in numerical dot notation, of the node or nodes that you do not want to poll. You can use IP address ranges and a wildcard character (‘*’) to specify multiple nodes.

- **StartDay**
  Specifies the day of the week that you want to start the polling off period. Specify the day by the first three letters, such as Sun, Mon, Tue, and so on. This field is not case-sensitive.
Start Time
Specifies the time of day that you want to start the polling off period. Specify the time in 24 hour time (colon separated).

End Day
Specifies the day of the week that you want to end the off period. Specify the day as described for StartDay.

EndTime
Specifies the time of day that you want to end the polling off period. Specify the time as described for StartTime.

PollTypes
Specifies the poll types that you want to turn off. Enter one or more of the following: p, c, d to specify ping, configuration, and discovery, respectively.

For example, add entries in the \usr\ov\conf\offperiods.conf file similar to the following:

16.21.144.140 Fri 23:00 Mon 6:00 pcd
16.21.[140-150].* Fri 23:00 Mon 6:00 pcd

After editing the \usr\ov\conf\offperiods.conf file, click on Apply or OK on the Polling Options dialog. The changes take effect immediately without stopping the daemons.

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 (called a demand poll). Normally, NetView 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, NetView 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 to create an SNMP Status entry using the Network Monitor Seed File Editor. Or, you may edit the netmon.seed file using a text editor; the prefix $ denotes an SNMP Status entry. Refer to the netmon.seed file for syntax information.
3. Explicitly, by specifying the P switch in the \usr\OV\conf\oid_to_type file against classes of devices.

Normally, a demand poll causes a PING to be sent to all of the instances of a node. If netmon is configured to use SNMP to poll for status, the demand poll will instead report the values of the ifAdminStatus and ifOperStatus for each interface of the node instead of sending PING requests to each interface. To use PING to poll for status on these nodes during a demand poll only, specify the switch -E in the \usr\ov\lrf\netmon.lrf file.

Configuring SNMP Nodes
You can make the following changes to the default configuration for SNMP nodes by selecting Options..SNMP from the NetView main menu:

• Change the status polling intervals.
• 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.

• Configure a specific node or a group of nodes to have different values than the default configuration.

• Configure a specific node or a group of nodes to have different values than the default configuration.

Configure SNMP Timeout Intervals

A timeout interval is the amount of time the management system waits for responses before retrying an SNMP request. To configure SNMP timeout intervals, select Options..SNMP from the NetView main menu.

If the timeout interval is long, there is less traffic generated. However, if your timeout interval is set too long, problem notification might not be timely.

Discovering Services on Nodes

Part of managing a network involves knowing which nodes provide important network services, such as DNS servers and web servers, and keeping track of the status of these services. The nvsniffer application provides the ability to discover and monitor these services, including the ability to use custom applications developed in-house.

The nvsniffer application uses a configuration file to govern which services to discover, which service SmartSets to create, which ports to test for a given service, and whether to use custom tests for discovering and/or status checking a service. (The configuration file is described in the subsection entitled Managing the nvsniffer.conf File). The default configuration file contains entries for common services of interest to most network managers and these services are declared present if the basic port test succeeds. The use of custom modules can greatly enhance the level of service capability on a node.

Service SmartSets are created by nvsniffer when nvsniffer is invoked against the configuration file where the services being tested are designated. Nodes become members of service SmartSets if the service test on the node succeeds.

For each service successfully discovered by nvsniffer, a service object is created and stored in the object database. The Node field that is specified in the nvsniffer configuration file entry for the given service is created and set to TRUE in the node object to which the service belongs; this automatically makes the node a member of the service SmartSet, if one is specified in the configuration file entry for that service.

Service objects are represented at the node’s interface level. The label assigned to a service object is the Service Label Name field from the corresponding nvsniffer configuration file entry. The status of each service is set each time nvsniffer is run (both Discover mode and Status mode keep the status of service objects up-to-date). Service objects that have been in the Critical state for the maximum allowed down time are automatically deleted from the object database. The corresponding node object’s boolean field is also deleted, effectively removing the node from the service SmartSet.

Managing the nvsniffer.conf File

The default nvsniffer configuration file is \usr\ov\conf\nvsniffer.conf, and it contains entries for some of the more common TCP services that nodes may have. You may add to this file, modify it, or create new configuration files that can be used by different incantations of nvsniffer. If a configuration file other than the default one is
to be used, 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\nv.log file, and the entry in which the error(s) occurred is ignored. Only one mode can be specified for a single invocation of nvsniffer: Discover mode or Status mode (specified as one of the command line arguments to nvsniffer). If -s for Status mode is not specified, the default mode of Discover mode is assumed. The format for each entry is: [Node Field][TCP Port,,][Service SmartSet][Service Label Name][Discover Test][Status Test][Discover Node Selection Criteria].

where:

[Node Field] is the name of the boolean field to create (if needed) in a node object based on the return status of a Discovery test. This field cannot be empty. [TCP Port,,] is a comma-separated list of the TCP port numbers to test on each selected node. Alternatively, this list can consist of any set of values (arguments) if custom test are specified and they require something other than TCP port number(s). This field can be empty ONLY if a custom test is specified in BOTH the Discover test AND Status test fields.

[Service SmartSet] is the name of the service SmartSet to create (if it does not already exist) during an nvsniffer Discover mode run. This field should be empty if no service SmartSet is to be created. For nvsniffer Status mode, this field is used as the node selection criteria. If this field is empty, then this entry of the file is ignored during an nvsniffer Status mode.

[Service Label Name] is the label name assigned to the service object created as a result of nvsniffer successfully discovering the service at the nodes being tested. The label name is also stored in the service SmartSet specified in the [Service SmartSet] field. This field can only be blank if the [Service SmartSet] field is left blank, and cannot be blank if a [Service SmartSet] field is specified.

[Discover Test] is the optional custom plug-in module to use to discover the service. This must be left empty if there is no custom discovery test. The name of a launchable application or an extension DLL with the NvServiceTest() API is what is specified here. The same launchable application or DLL can be specified as that specified in the [Status Test] field. For nvsniffer Status mode, this field is not used.

[Status Test] is the optional custom plug-in module to be used to status check the already discovered service. This must be left empty if there is no custom status test. The name of a launchable application or an extension DLL with the NvServiceTest() API is what is specified here. The same launchable application or extension DLL with the NvServiceTest() API can be specified as that specified in the [Discover Test] field. For nvsniffer Discover mode, this field is not used.

[Discover Node Selection Criteria] specifies which nodes to test during a Discover mode run. It can be either * for all IP nodes, or the name of ANY existing SmartSet. This field is not used during an nvsniffer Status mode run. All entries in the configuration file that specify * are grouped together so that all discovery tests can be done to an IP node all at once. Entries that specify a SmartSet for this field are processed individually; the node worklist is re-loaded with the members of the specified SmartSet for each test.

nvsniffer Discover Mode

Discover mode is the default mode for nvsniffer. When nvsniffer is launched in Discover Mode, nvsniffer tests for the existence of each service against the nodes
specified in the Node Discovery Selection field of each entry in the configuration file. The default configuration file is located in \usr\ov\conf\nvsniffer.conf. The default test consists of a raw port test of the enumerated TCP ports or, if specified in the Discover Test field of the configuration file entry, the custom discovery test is used instead of a raw port test. See [Using Custom Plug-In Modules for Discovery and Status Checking].

nvsniffer Status Mode

Launching nvsniffer in Status Mode ("-s" switch) causes known service on the nodes that are members of Service SmartSets to be status-checked, as specified in the configuration file being used. If specified, the custom Status Test or the NvServiceTest() API in the custom extension DLL is invoked. For more information, see [Using Custom Plug-In Modules for Discovery and Status Checking]. Otherwise a raw port test against the enumerated TCP port numbers specified in the TCP Port field is performed. Node selection criteria for nvsniffer Status Mode consists of the nodes that are members of the Service SmartSet specified in each configuration file entry. Service SmartSets are created by nvsniffer during a Discover Mode run.

Automatic ReInvocation of nvsniffer

To enable automatic relaunching of nvsniffer service discovery or service status testing, specify the -r <minute> switch when invoking nvsniffer. This causes nvsniffer to automatically reschedule itself using the current settings (command switches) to run again after the specified number of minutes. The valid range for minutes is 5..44640 (the maximum value is 31 days).

To view the settings of a scheduled nvsniffer AT job, enter “at” at the command prompt. Note that the Windows Scheduler Service must be configured and started for this feature to work.

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 an nvsniffer configuration file entry. The same plug-in module can be used for both tests. Custom plug-in tests may consist of any thread-safe, re-entrant launchable script or executable file, or it may consist of a thread-safe, re-entrant extension DLL that contains the NvServiceTest() API.

Refer to \usr\ov\doc\nvsniffer.readme for details on how to create and use custom plug-in modules.

Logging the Progress of nvsniffer

To capture the progress of an nvsniffer run, specify the -l <log_file_name> switch to log all progress information. Errors are logged in the standard error log file \usr\ov\log\nv.log and are output at the command prompt if the nvsniffer application was launched from a command prompt.

Testing One Node

To test a single node for the services specified in a configuration file, use the -n <node_name> switch to specify the node to test. This switch can be used for either an nvsniffer Discover mode or Status mode run. When specified for a Discover mode run, the service entries in the specified configuration file that have a Discover Node Selection Criteria of * (for all nodes) are tested, and any entry specifying a SmartSet for the Discover Node Selection Criteria in which the specified node
belongs will be tested. For a Status mode run, the services that are status checked
are those in which the node is currently a member of the corresponding Service
SmartSet.

**Force Test of Services on Critical Node**

Normally, nvsniffer will ignore nodes that have an IP Status of Critical. To override
the critical status and force a test of services, specify the -o switch.

**Managing Network Devices through the Web**

Certain network devices can be managed through World Wide Web pages. You can
configure NetView to access the management pages directly from the NetView
submap. Select the device symbol, then click on the Management Page item on
the pop-up menu. Your default web browser will be invoked with the default home
page on that device.

You can also invoke the management page from the Web client by selecting the
symbol representing the device and clicking on the Management part of the
symbol. Note that only symbols representing objects with the ManagementURL field
set will have this management section.

To use this information, specify the management page URL for each object that can
be managed through the Web. Manually select Object Properties or automatically,
use SNMP System Object IDs. Normally, you would use the automatic
configuration, and if necessary, use the manual configuration if the home page URL
was not appropriate for a particular device.

**Manual Configuration**

Select the symbol on the map representing the managed device and click Object
Properties in the pop-up menu. Select the Other tab page and select LANMAN
from the pull-down box. Check the field isHTTPSupported and isHTTPManaged.
Type the appropriate URL for the ManagementURL field.

**Automatic Configuration**

If you know that all your devices with a particular SNMP SysOID can be managed
through the Web, you can modify the \usr\ov\conf\oid_to_type file to have the
associated fields filled in automatically upon discovery. See the comments in this
file for the W flag and syntax details, but basically, you add the W flag to the line
containing the SysOID. This will cause the home page URL
(http://myrouter.tivoli.com for myrouter.tivoli.com) to be put in the ManagementURL
field for all devices with that SysOID.
Chapter 9. Resolving Network Problems

The NetView program enables you to perform fault management tasks to locate and resolve problems that occur in networks managed by the NetView program.

Using the NetView program's fault management functions is a two-part process. The first part involves continuous routine monitoring of the networks managed by the NetView program. Monitoring a network's behavior gives you a historical record of its performance, which can be helpful if you need to re-create a particular network configuration after discovering a widespread problem. You can also monitor the network's real-time performance to track developing trends and lessen the risk of major problems.

The second part requires active problem resolution. When the NetView program displays problem areas, you can take steps to determine the source of the problem and resolve it. During this phase, you might use some of the historical information you collected by monitoring the network. You might also continue realtime monitoring to ensure that your problem resolution is having the desired effect.

This chapter shows you ways to use the NetView program to find and solve problems with your network. The following topics are covered in this chapter:

- “Gathering Information about Your Network” on page 102
- “Classifying Network Problems” on page 103
- “Diagnosing Network Connectivity Problems” on page 107
- “Diagnosing Network Performance Problems” on page 107
- “Diagnosing Network Service Problems” on page 110

If you need help diagnosing and solving problems with the NetView program itself, see “Chapter 10. Diagnosing and Solving NetView Problems” on page 113.

Gathering Information about Your Network

The NetView program enables you to gather and save information about your network that can help you track and resolve problems. This section describes some ways to gather network information.

Taking Snapshots of a Map

It is important to have a record of the status of a map in case it needs to be re-created. Taking snapshots at regular intervals can help you maintain a history file to consult when you need to make changes or resolve a problem. You can also take snapshots between regularly scheduled intervals if they are needed. Snapshots are read-only and are not updated.

You can list all the snapshots you have saved by selecting Window..Maps or by selecting File..Open. When you select File..Open, a table of all maps and any snapshots for the open map is displayed. The Permissions column in the table tells whether each entry is a map or a snapshot.

Taking Snapshots at Regular Intervals

Taking snapshots of the topology database at regular intervals can help you maintain a comprehensive historical record of database changes. Use the Windows at command to set up periodic execution of the ovmapsnap command.
For example, the following `at` command causes a snapshot to be taken at 1:00 a.m. every Saturday:

```
at 1:00 /every:s "ovmapsnap -c \"Automatic snapshot\" " "
```

In this example, the NetView program sends the message `Automatic snapshot` to the user.

To use the `at` command, you must have the Windows Schedule Service running. Select `Control Panel..Services` to see if the Schedule Service is started.

After you set up an `at` command, it automatically creates snapshots until it is stopped. Over time, the map snapshot file can grow unmanageably large.

- To delete older map snapshots, use the `ovmapsnap -d` command or regularly selecting `File..Delete` to delete map snapshots that are no longer needed.
- To delete only selected map snapshots, use the `ovmapsnap -d -n <snapshot name> -m <mapname>` command.
- To delete the oldest entry, use the `ovmapsnap -d -f -m <mapname>` command. Refer to the `Tivoli NetView for Windows Programmer's Reference` for additional information about the `ovmapsnap` command. To get information about the `at` command, double-click on the Windows Help icon in the Program Manager, then select MS DOS Command Reference Help from the Help contents.

**Taking Snapshots as Needed**

To take a snapshot of the open map, select `File..Save As` from the NetView main menu, then select `Snapshot` in the Save As field in the dialog box that appears. You can display the current topology map and topology database with a snapshot by selecting `File..Open`, then selecting a snapshot from the list of maps and snapshots.

**Polling the Network**

Generally, agents in the network are regularly polled to update the status of the NetView objects. The information obtained by this regular polling is used to update the topology map.

- To collect network status at regular intervals, use the `Options..Polling` command.
- To collect status information between scheduled polling intervals, select the node or nodes whose status you want to know, then select `Test..Demand Poll` from the NetView main menu. If a node suddenly goes down or if you suspect problems with network traffic to or from the node, a demand poll can help you pinpoint the problem area.

Regardless of whether polling is conducted at regular intervals or on demand, the polled data includes the following information:

- Object status
- Network topology changes, including new nodes
- Configuration changes
- Surpassed thresholds that indicate a set limit was reached or exceeded

You can also use the MIB Data Collector to collect selected MIB variables from selected nodes at regular intervals for trend analysis. The MIB Data Collector generates an error notification when the thresholds you specify are exceeded. You can dump the collected data to a spreadsheet program to help you chart trends. Use the `snmpodump` command to dump the data. Refer to the `Tivoli NetView for Windows Programmer's Reference` for more information about this command.
Monitoring the Network

To help anticipate problems, you can monitor the network by using a variety of tools that enable you to view the network from different perspectives. The following list presents some monitoring options you might want to consider using:

- Establish baselines for normal network performance.
  To establish a baseline, select the statistics you want to monitor and keep track of them over a one- or two-week period. Save the graphed results for comparison with graphs of those same statistics that you collect while you are determining the cause of a problem.

- Build applications to monitor trends.
  Use the MIB Tool Builder to build an application that queries and displays a device’s current MIB values. The data can be displayed as text or a graph and can be saved to a file so you can look at it later.

- Set up thresholds for monitored MIB values.
  After you analyze the trends, select Tools..MIB..Collect Data to set a threshold on each MIB object. After a threshold is set, an event is generated and the event display is updated when the threshold is exceeded. You can graph the results and save them to a file for later use.

- Fine-tune the thresholds.
  Monitor the threshold for a period of time to ensure that you have set up the appropriate values. If you are receiving several threshold events a day without any loss in network performance, you might want to change the threshold and re-arm values.

Viewing Network Events

Check the Event Browser for events that are problems or could become problems. The NetView program records network events and creates the following event categories shown in Table 17.

<table>
<thead>
<tr>
<th>Events Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threshold Events</td>
<td>A threshold was exceeded.</td>
</tr>
<tr>
<td>Network Topology Events</td>
<td>An object or an interface was added or deleted by an application.</td>
</tr>
<tr>
<td>Error Events</td>
<td>An inconsistent or unexpected behavior occurred.</td>
</tr>
<tr>
<td>Status Events</td>
<td>An object or an interface status changed to up or down, because the object or interface started or stopped responding to ICMP echo requests.</td>
</tr>
<tr>
<td>Node Configuration Events</td>
<td>A node’s configuration changed.</td>
</tr>
<tr>
<td>Application Alert Events</td>
<td>A NetView application generated an alarm or alert.</td>
</tr>
</tbody>
</table>

To see the events in each category, select Monitor..Events from the main menu. To see the events of a given severity, select Filter..Set, then create a filter for the severity.

From the main menu, select Options..Events..Trap Settings to define the format of the event log messages and the actions to take when the management system receives an event. This command lets you customize the management system and
automate some of the fault management tasks. For more information on configuring events, see “Configuring Events” in the online help.

You can also use the MIB Tool Builder to write an application that can take specific action when a threshold event occurs. For example, the application might:
- Display an informational or warning message in a dialog box.
- Activate a fault recovery script.
- Activate a hand-held beeper.

For information about building a MIB application, see “Working with MIB Applications” on page 80.

**Displaying Critical Objects**

You might want to display all network resources below a particular level. If network resources are down, the symbols are red. You can use the Critical Objects Window to determine if a problem exists or is developing.

Select a network symbol on the currently displayed submap. Next, select View..Critical Objects to display, in a new submap, the lowest-level underlying critical resources for the submap level where you are currently working. From the Critical Objects window, you can select one of the failed interfaces and double-click on it to display the selected symbol and its higher-level connections.

You should wait until most of your network’s discovery is completed before you use the Critical Objects window. Otherwise, the resulting display might not give you an accurate or helpful picture of critical resources.

The Critical Objects display is automatically updated if the status of one of the resources changes. This change in status might be the result of the interface’s becoming active again or because you selected the interface or the network node to which it is attached, then selected Object..Acknowledge from the NetView main menu. In addition, resources whose status changes from Up to Down while the Critical Objects window is active are added to the display.

Any resources that you might add to the map during the current session are not displayed even if their status changes from Up to Down. To display these newly added resources, you must exit the current Critical Objects window and select View..Critical Objects again.

**Displaying Node Information**

Select Monitor..Network to retrieve and display specific information about a selected node. This information can include the following:
- A list of network interfaces, including TCP/IP and physical (link) addresses
- The routing table for the node
- A list of TCP/IP services available for the node
- A list of currently active TCP/IP connections

**Classifying Network Problems**

Network problems are generally in one of the following categories:
- Device errors caused by a hardware or software failure in a specific network node
- Connectivity errors caused by the lack or failure of a connection between devices, usually as the result of a device error
For example, the failure of a LAN interface card on a gateway between networks is a device error, which can lead to a connectivity error when internetwork communication is attempted. A network problem might display symptoms related to performance, connectivity, or network services. How you diagnose and resolve the problem depends on the type of symptom that is displayed.

---

**Diagnosing Network Connectivity Problems**

A network connectivity problem exists when a user cannot contact a system on the local network or on another network.

**Symptoms of Network Connectivity Problems**

A network connectivity problem might be revealed by one of the following symptoms:

- The user cannot reach a system that could previously be contacted by using, for example, the `ftp` command.
- The user receives an error that the connection has timed out. The timeout could be caused by any of the following situations:
  - A system is down.
  - There is a routing problem.
  - The default gateway has a problem.
  - Poor network performance results in packets not getting through in time.
- The user receives an error that the remote system could not be found. Such an error could be caused by one of the following situations:
  - The system is powered off.
  - The system no longer exists on the network.
  - The system has a hostname-to-IP mapping problem and does not recognize the host name.
  - The gateway does not have the remote system in its routing table.

**Testing Connectivity**

Use the following NetView commands and features to test your network’s connectivity:

- Select **Test..Connectivity** to test the TCP/IP and SNMP protocols. This selection polls the device at the TCP/IP and SNMP layers to verify that it is connected to the management system.
- Select **Test..Ping** to determine the number of packets sent and lost, and the round-trip times between the manager and a remote node. This command verifies connectivity at the IP level.
- Examine patterns in the status colors to determine connectivity problems. For example, if a connector and all objects on one side of it are down (indicated by the color red), the connector is probably the source of the problem.
- QuickTest is an enhanced ping tool that polls all managed interfaces and quickly determines the complete status of the node. QuickTest displays the ifAdminStatus and ifOperStatus of each interface for nodes that use SNMP to poll for status. For all other nodes, QuickTest displays the ping result for each interface. If the status of an interface has changed, the map is updated and an event sent.
- QuickTest is available from the NetView console and NetView Web Console by selecting **Test+QuickTest** or **Test+QuickTest Critical.** You must have one or more nodes selected to access the QuickTest menus. The Test+QuickTest Critical menu item only polls interfaces that are currently marked as down. The Test+QuickTest menu item polls all interfaces, regardless of their current status.
You may also run QuickTest from the command line. The QuickTest command has the following syntax:

```
quicktest -n nodename [-z] [-P] [-F]
```

where `-n nodename` specifies the node to be polled and is required.

Other options include:

- `-z` Only poll down interfaces
- `-P` Using ping instead of SNMP requests to poll SNMP status nodes
- `-F` Send an event for all polled interfaces, regardless of whether there is a status change. Normally, an event is only sent when the status has changed.

### Locating Routes between Nodes

To keep certain parts of the network running, you might need to locate an alternate route between two nodes so all traffic between them can bypass another route that is having problems. To locate specific IP routes between two selected nodes, select Test..Locate Route.

This selection highlights both nodes and all IP-addressable objects and connections between them. The command also lists such information as the source node and node address and the intervening gateways and gateway addresses.

### Gathering Connectivity Information

Gather and analyze the following information to help diagnose connectivity problems:

- Select Monitor..Network..Routing Table to obtain gateway routing table information for a remote SNMP node.
- Select Monitor..Network..ARP Cache to obtain address translation table information for a remote SNMP node.
- Select Monitor..Network..Addresses to determine IP and link addresses for a remote SNMP node.
- Select Monitor..Network..IP Services to determine network services available on a remote SNMP node.

### Checking Connectivity within the Same Network

Use the following procedure when a user is having problems connecting from one system to another system within the same network.

**Prerequisites:**

- To connect to a node through Telnet, the remote node must have TCP Services running.
- To use the ARP Cache operation, the selected node must support SNMP.

In Figure 26 on page 105, a user cannot connect from System A to System B on the same network.
1. Look at the status color of System B on the map. If it is red, System B is down. If it is not red, go to Step 2.

2. Check to see if System A and System B are operational.

3. Select **Test..Ping** to send a ping from the management system to System A and System B.
   - If the ping to either system fails, check to see if the system is down.
   - If the ping to both systems is successful, go to Step 4.

4. Determine if there is a hostname-to-IP address mapping problem.
   - On the map, select System A, then select **Tools..Telnet** and login to System A. After you have logged in, ping System B. If the ping fails, go to Step 5.
   - If the ping works, you might have a hostname-to-IP address mapping problem. To verify and fix the problem, do the following:
     a. Look in the hosts file for the hostname of System B, as defined to System A. If System A is a UNIX workstation, the hosts file is /etc/hosts. If System A is a Windows PC, the hosts file is \winnt\system32\drivers\etc\hosts, where winnt is the release directory for the release of Windows that is running on the management system.
     b. Telnet to System B, and then ping System A. Look in the System B hosts file for the hostname of System B. It should be the same as System B’s name in the System A hosts file.

5. Determine whether there is an IP-address-to-link-address mapping problem.
   - On the map, select the management system and System A.
   - Select **Monitor..Network..ARP Cache** to view the ARP cache tables of the management system and System A.
   - Locate the IP address of System B in both tables and compare the link address for System B.
     - If System A’s link address for System B is not the same as the management system’s link address for System B, change System B’s link address on System A.
     - If the link addresses are the same, you might not have a connectivity problem, but rather a performance or network service problem. For more information, see [Diagnosing Network Performance Problems](#) on page 107 and [Diagnosing Network Service Problems](#) on page 110.

**Checking Connectivity across Networks**

Use the following procedure when a user is having connectivity problems from a system on one network to a system on a different network.
Prerequisites:

To use the Locate Route command:

- Select two nodes: a source node and a destination node.
- All nodes along the route, except the destination node, must be running SNMP.
- All gateways between the selected nodes must have their community names correctly set.

In Figure 27, a user is trying to connect from System B on Network 1 to System C on Network 2. The path between the two networks includes two gateways, G1 and G2.

1. If System C’s status on the map is red, System C is down. If it is not red, go to Step 2.
   
2. Check connectivity to System C.
   
   Select **Test..Ping** to send a ping from the management system to System C. If the ping is successful, go to Step 3. If the ping fails, take the following action:
   
   a. Send another ping from the management system to G2.
   
   b. If the second ping is successful, go to Step 3. If the second ping fails, G2 is down.
   
3. Check connectivity to System B.
   
   Select **Test..Ping** to do a ping from the management system to System B. If the ping is successful, go to Step 4. If the ping fails, take the following action:
   
   a. Send another ping from the management system to G1.
   
   b. If the second ping is successful, go to Step 4. If the ping fails, G1 is down.
   
4. Find the route between the systems and all intermediate objects between them.
   
   On the map, select System B, then System C. Select **Test..Locate Route**.
   
5. Look at the information in the dialog box to see if the path is correct. For example, if the route leads to G3 rather than G1 and G2, the routing table for System B is incorrect.
If the route is on a single submap, it is simple to follow. If the route is not completely visible from a single submap, you can open other submaps along the path. However, you can also select Test..Locate Route to verify the Next Hop host names and IP addresses.

6. Look at the map to verify that all nodes along the correct path are operating. A node with a yellow or red status indicates that the node either has a problem or is not operating. If a node is yellow or red, diagnose the problem on that node.

7. If the route is correct and everything is operating along the path, the problem might not be related to connectivity, but rather to performance or network service. For more information, refer to Diagnosing Network Performance Problems and Diagnosing Network Service Problems on page 110.

Diagnosing Network Performance Problems

A network performance problem might be revealed by one of the following conditions:

- Users experience slow system response while running applications or during communications across the network. This could be caused by a traffic overload on the network or an unhealthy device, such as a gateway.
- Users experience a loss of data during transmission. A network or device might be overloaded, causing a timeout.

Prerequisites:

- The originating node can reach the destination node without losing a connection.
- To use the Locate Route command:
  - Select a source node and a destination node.
  - Ensure that the community names for all gateways between the selected nodes are correct.
- To monitor interfaces or interface traffic on a selected node, the node must support SNMP.

In Figure 28 on page 108, the user was experiencing unusually slow response time trying to communicate from System B on Network 1 to System C on Network 2. Suppose that, from the management system, the response time on Network 3, where System E resides, is normal. In this example:

- System A and System B are both on Network 1 (G1)
- System C is on Network 2 (G2)
- System E (which has relatively low usage) is on Network 3 (G3)
- Two gateways exist between System B and System C
- The networks are Ethernet networks
1. Select System B and System C. Then select **Test..Locate Route** to find the route between the systems and all intermediate objects between them. In this example, the route is B to G1, G1 to G2, and G2 to System C.

2. Select System A and connect to it by selecting **Tools..Telnet**. From System A, ping System C. Then select System B and connect to it by selecting **Tools..Telnet**. From System B, send pings to Systems C, D, and E. Record the response times on these pings, and then diagnose the problem as follows:
   - If the response time from B to C, B to D, and A to C is slower than the response time from B to E, you have a network problem. Go to Step 3.
   - If the response time from B to C and A to C is slower than the response time from B to D and B to E, the central processing unit (CPU) for System C is overloaded. Select System C and log into it by selecting **Tools..Telnet**. Then, run a local CPU monitoring utility to verify the CPU overload for System C. Next, reduce the CPU load for System C, for example, by stopping some processes or having users log off the system.
   - If the response time from B to C, B to E, and B to D is similar but slower than the response time from A to C, the CPU for System B is overloaded. Select System B and log into it by selecting **Tools..Telnet**. Then run a local CPU monitoring utility to verify the CPU overload for System B. Next, reduce the CPU load for System B, for example, by stopping some processes or having users log off the system.

3. To locate the network problem, select **Tools..Telnet** and **Test..Ping** to do the following:
   - Log into System B and ping G1.
   - Log into System G1 and ping G2.
   - Log into System G2 and ping C.

   Record the response times on these pings, and then diagnose the problem as follows:
   - If the response time from B to G1 is faster than the response time from G1 to G2 and G2 to C, check the traffic on G2. Go to Step 4.
   - If the response time from G2 to C is faster than the response time from B to G1 and G1 to G2, check the traffic on G1. Go to Step 4.
   - If the response time from G1 to G2 is slower than the response time from B to G1 and G2 to C, the problem resides between G1 and G2. You might need to set up a different route, subnetwork, or gateway on your network.

4. Select the gateway with the problem. Select **Monitor..Network..Interfaces** to see trends of the interface traffic displayed in a table. Select **Monitor..Network..Interfaces** to see a snapshot of the interface data displayed...
in a table. If input and output data on interface traffic is abnormally high, you might have a routing problem, an overloaded interface, or an overloaded gateway. In this case, select **Test..Locate Route** to find alternate routes and reroute network traffic, accordingly.

### Monitoring Performance and Traffic

If you suspect performance problems, you can monitor and graph certain statistics to help isolate problems. First, verify that the SNMP agent is running for the nodes you want to check. Then select the following menu options to observe traffic patterns on the nodes, peak values, and averages over given intervals:

- **Monitor..Network..Interfaces**
- **Tools..MIB..Browser**
- **Tools..MIB..Collect Data**

To monitor performance and traffic on other vendor devices, or to monitor MIB variables other than MIB-2, create MIB applications by selecting **Tools..MIB..Tool Builder**. Status polling generates the heaviest amount of network traffic. You can monitor the traffic generated by the NetView program by selecting **Monitor..Network..Interfaces**. Use this command several times and observe the differences in the number of packets sent and received.

### Gathering Performance Information

Gather and analyze the following information to diagnose performance problems:

**Note:** The information that can be gathered for monitoring and graphing depends on the vendor MIBs.

- To detect changes in the status of interfaces, check the node color (indicating the status) and select **Object..Object Properties** to see the status of a specific interface.
- To check for errors in incoming and outgoing packets and errors for each connected interface for a remote SNMP node, select **Monitor..Network..Interfaces**.
- To determine the round-trip times between nodes, select **Test..Ping**.
- To analyze trends in collected MIB data, select **Tools..MIB..Collect Data**.

### Regulating NetView Traffic

To regulate traffic generated by the NetView program, follow these steps:

1. Adjust polling intervals by selecting **Options..Polling**. Note that reducing the amount of polling traffic by lengthening polling intervals can delay real-time map updates, resulting in a less accurate map.
2. Unmanage nodes by selecting **Object..Unmanage**. When you monitor only what needs to be monitored, event traffic decreases. Also, when you unmanage an object, the amount of management information available for unmanaged nodes decreases.
3. Adjust the amount of MIB data collected by selecting **Tools..MIB..Collect Data**. Do this by using one of the following methods:
   - Stop collecting MIB data on some of the MIB objects that are configured to collect data by changing the status to Suspend.
   - Modify the details of the configured MIB objects by deleting nodes, excluding nodes, or changing polling intervals.
Regulating netmon Broadcast Traffic

The netmon daemon generates some network broadcast traffic as a result of polling to discover network objects and to request status updates from agents. The amount of traffic generated by the netmon daemon can be influenced by any of the following considerations:

- The number of nodes that are up and the number of nodes that are down
- The number of nodes that are on the same subnetwork
- The number of nodes on other subnetworks
- The size of the permanent ARP cache
- The probability that a particular IP address can be found in the permanent ARP cache

Broadcast traffic generated by the default netmon command (no options) is dependent on the following:

- For nodes that are up, but not on the same subnetwork where the management system is located, netmon generates just one Internet Control Message Protocol (ICMP) echo request packet and receives one response. In this case, an ARP Request is not generated.
- Because the management system is on a different subnetwork, a request passes through a gateway, which always has a large ARP cache. The netmon daemon knows the gateway’s link-level address and does not have to broadcast an ARP Request to that gateway, so it sends only one ICMP echo request through the gateway and gets back one response.
- For devices that are on the same subnetwork where the management system is located, the netmon command might generate only an ICMP echo request. In addition, it might generate one ARP Request. Because the nodes are on the same subnetwork, netmon does not have to go through a gateway. Broadcasting an ARP Request depends on whether or not the netmon daemon’s ARP cache contains the gateway’s link-level address.
- The netmon daemon generates an ARP Request based on whether the IP address being polled is still in the ARP cache of that machine.
  The presence of the IP address in the ARP cache is affected by how many nodes are on the subnetwork and by how much physical memory is in the system. The system automatically adjusts the size of its ARP table.

The netmon daemon explicitly pings every IP address by default, which causes an increase in network traffic. Explicitly pinging every IP address is most useful when there are only a few SNMP agents on the subnet. To turn off explicitly pinging every IP address, select Options..Server Setup..Daemons..Netmon Daemon, then click on Discover using Spray Ping.

Diagnosing Network Service Problems

In a network service problem, a user reaches System B from System A, but either the connection from System B is refused or the command that is executed from System A to System B is not accepted.

A network service problem might be revealed by one of the following conditions:

- After connecting to a remote system, the user receives an error that the remote system does not accept the command that was initiated. This could be a service problem related to security.
• After connecting to a remote system, the user receives an error that the command sent to the remote system was not recognized. For example, System B does not allow login for an ftp command from System A. This could mean the service might not be installed or configured.

Prerequisites:
• To view the IP/TCP/SNMP protocols and services available on a selected node, the node must support SNMP.
• The originating node can reach the destination node without losing a connection.
• You must have privileges on the system you are trying to configure. For UNIX systems, you must be root. For Windows systems, you must be a member of the Administrators group.

In Figure 29, a user tries to use a network service, such as the ftp command, from System A to System B and receives an error message.

![Figure 29. Diagnosing Network Service Problems](image)

Perform the following steps to diagnose this network service problem:

1. Select Test..Connectivity on both System A and System B to ensure that they support the network protocols and that the protocols are working. If there are problems with the protocol, an error will appear in the message field.

2. If the nodes support SNMP, select Monitor..Network..IP Services to ensure that they support the service in question, such as FTP.

3. If the nodes provide the service, go to Step 5. If they do not provide the service, check the file system on the two systems to see if the software for the service, such as Telnet or FTP, is installed.

4. If the service is not installed, install the software on the system. If the service software is installed, configure the service on the system. Go to Step 5.

5. Check service security on System B. Ensure that the IP address of System A is set for allow access and not excluded by a deny access setting in the security files for the service in question.

6. Request the service again, from System A to System B, to verify that the problem is fixed.
Chapter 10. Diagnosing and Solving NetView Problems

This chapter helps you determine whether a problem is with the NetView program itself or with your network. It describes some common NetView problems and tells how to solve them and gives tips on how to avoid problems with the NetView program. The topics covered in this chapter are:

- “Determining the Problem”
- “IP Discovery and Layout Problems” on page 119
- “Diagnosing Problems with Topology, Discovery, and Database Daemons” on page 126
- “Diagnosing Problems with Event and Trap Processing Daemons” on page 131
- “Using Daemon Tracing” on page 133
- “Diagnosing Problems with the NetView Program” on page 134

Refer to “NetView Administration” in the online help for more information about the NetView setup options.

Determining the Problem

To diagnose a problem you have encountered while using the NetView program, you must determine whether the problem is with the NetView program or the network. Be sure the TCP/IP and SNMP network services are running correctly and check that your physical network (Ethernet, token-ring) is operating correctly by pinging another node.

To determine whether the problem is with the NetView program rather than with a managed node, run the NetView program from a different node. If the results are the same, the NetView program is probably at fault. If the NetView program fails to collect data from a remote node, realize that the amount of information that the NetView program can collect from a remote node depends on:

- Whether the node supports SNMP and is configured to send data to the manager through SNMP
- Whether the node supports a partial, full, or extended (nonstandard) MIB
- Whether the node is a TCP/IP node
- The node’s MIB characteristics

Identifying Problems

If you receive error messages while installing or running the NetView program, make a note of the message before closing the message dialog box. Also, if the NetView program operates in an unexpected way (for example, if response time is too long or if an unexpected response is received), make a note of the action you selected and the NetView program’s response. This information can help you determine:

- What is affected? Is it a device, a Windows process, or the NetView program?
- What part of the NetView program is affected? Is it all operations or just specific daemons or processes?

Installation messages are logged in the \<winnt\>\system32\nvinstall.log file. You can see this log by selecting the Installation Log icon from the NetView program group. Errors received while running the NetView program are logged in the \usr\ov\log\nv.log file.
The network manager gets the most information from an SNMP node and less information from non-SNMP TCP/IP nodes. If a problem suddenly develops while you are using the NetView program, review your most recent actions to eliminate yourself as the problem cause, and ask these questions:

- **What changed?**
  Did the configuration change (hardware, software, or security), or did the system utilization change immediately before the problem occurred? When you plan to change a configuration, make a note of the existing setup in case you have to restore it.

- **How often does the problem occur?**
  Is the problem consistent or intermittent? Intermittent problems are usually not caused by operator error.

- **What was the last action performed?**
  When the problem occurred, what operation or command was selected, and what data was sent or received?

### Installation Problems

When installing the NetView program, it is critical that you are connected to an operational TCP/IP network. As a network manager, the NetView program depends on access to the network, even during installation, and you will see a warning message if you are not connected to the network.

Follow this checklist for installation problems:

- Check the prerequisites for running the NetView program. The following software components must be installed, configured, and operational:
  - Windows operating system
  - SNMP agent
  - TCP/IP

- Ensure there is enough disk space where the NetView program installs. Use the Windows File Manager or the directory command to check available disk space.

- Ensure that there is enough paging space. Use the Windows System icon in the Control Panel group. Click on the Virtual Memory button. The suggested minimum paging space is 120 megabytes.

### Runtime Problems

To diagnose problems with the NetView program that occur while the program is running, you need to know the status of the NetView daemons. After you find out that a daemon is not running, you can restart it and any other daemons on which it depends. The following topics describe how to check daemon status and restart a daemon that is not running properly.

#### Checking Daemon Status

Daemons are processes that run in the background. When the NetView program is running properly, several daemons are running in the background. You can see the status of daemon processes by taking any of the following actions:

- Double-click the **Daemon Status** in the NetView program group.
  The Daemon Status icon reports the current status of each daemon in a display window.

- Enter the **ovstatus** command in a command prompt window to check the status of all currently running daemons. For more information, see the ovstatus reference page in the *Tivoli NetView for Windows Programmer’s Reference*.

To check the status of certain daemons, enter the following command:
ovstatus [-d] [daemon_names]

• Select Options..Server Setup from the NetView main menu, and then select the Daemons tab page of the NetView Server Setup dialog box.

The Daemons tab page of the NetView Server Setup dialog box lists all of the NetView daemons, and gives status information on each, such as whether a daemon is running and the last message received from the daemon.

The status output you receive when you double-click on the Daemon Status icon or enter the ovstatus command is similar to this example:

object manager name: owwdb
behavior: OVs_WELL_BEHAVED
state: RUNNING
PID: 4115
last message: Initialization complete
exit status: -

If the state for any daemon is anything but RUNNING, you will have to restart that daemon. See [Starting Daemons].

If the process management daemon, ovspmd, was started with the -V option, this status report is much more extensive, including everything the ovspmd daemon knows about the daemons.

Checking the NetView Console
In addition to the daemons, the NetView console must also be running for the NetView program to work. Some other processes run only if the NetView console is active. These include nmPolling and ipmap.

To verify that the NetView console is running, double-click on the Process Viewer icon in the NetView program group and make sure that netview.exe is in the list of active processes. If it is not running, click Start..Program..NetView.. NetView Console.

Starting Daemons
You can start the NetView daemons in any of the following ways:

• Double-click the Start Daemons icon in the NetView program group.

  The Start Daemons icon starts all the daemons that are configured to start in the \usr\ov\conf\ovsuf file. As each daemon is started, its status is displayed in a dialog box. The dialog box displays if the daemon started or failed to start. To make sure all the daemons have started and are running properly, double-click the Daemon Status icon in the NetView program group.

• Enter the ovstart command in a command prompt window.
  − If you enter the ovstart command with no arguments, it starts all the daemons that are configured to start in the \usr\ov\conf\ovsuf file.
  − To start only selected daemons, enter the ovstart command with arguments that specify which daemons to start.

The ovstart command does not exit until all the daemons it has tried to start have either started or definitely failed to start. By default, it creates no output unless a daemon fails to start.

For more information, refer to the ovstart reference page in the Tivoli NetView for Windows Programmer's Reference.

• Select Options..Server Setup from the NetView main menu, and then select the Daemons tab page of the NetView Setup dialog box.
– On the Daemons tab page, select the **Start All** button to start all the daemons listed on this tab page.

– To start a single daemon, select the daemon in the list and then click **Start**.

### If a Daemon Is Not Running

If a daemon is not running, the problem might be one of the following:

- The **ovspmd** daemon configuration is damaged. If this is the case, see [**ovspmd Daemon**](page 131).

- The daemon executable has been removed or moved from its expected location. The `\usr\ov\conf\ovsuf` file contains the path name to each daemon. By default, all are in `\usr\ov\bin`.

When the status report indicates that a daemon is not running, double-click on the **Stop Daemons** icon in the NetView program group to stop all daemons. To restart the daemons, double-click on the **Start Daemons** icon, or `ovstart -v` (verbose) to create a report for each daemon that indicates whether it is started.

### If You Receive a Startup Error Message

When you double-click on Start Daemons, you may receive an error message from the ovspmd daemon, such as:

```
ovspmd: \usr\ov\conf\ovsuf: object manager ovtopmd not registered
```

This message may mean that the ovsf configuration file has been damaged. To re-create the file, follow these steps:

1. Delete the `\usr\ov\conf\ovsuf` file using the delete command.
2. Change directories to `\usr\ov\lrf`.
3. Enter the `\usr\ov\bin\ovaddobj <daemon.lrf>` command. The `<daemon.lrf>` field is any daemon in the lrf directory. This command reregisters the daemon you specify.
4. Any daemon not reregistered must be stopped. Enter the `\usr\ov\bin\ovstop daemon_name` command. The daemon_name is the name of the daemon you want to stop.
5. To restart the daemons and verify that they are starting properly, double-click on the **Start Daemons** icon in the NetView program group.

If any daemons have still not started, you might have another ovspmd daemon problem. See [**A Daemon Is Not Running**](page 131), and determine whether you can identify and resolve the problem.

### Registering Daemons

You can use the `ovaddobj` command to register the information necessary to start daemons with the ovspmd process management daemon. Daemons are configured by the `ovaddobj` command from information in Local Registration Files (.lrf), and are identified by a combination of host name and daemon name.

To change any other information about the daemon, use the `ovdelobj` command to delete the lrf, then the `ovaddobj` command to add a new lrf. Refer to the ovaddobj, ovdelobj, and lrf reference pages in the *Tivoli NetView for Windows Programmer's Reference*.

### Verifying Local Registration Files (LRFs)

The `ovaddobj` and `ovdelobj` commands read the local registration files (LRFs) to register or deregister an agent. The `.lrf` file contains information needed to start the agent program as a daemon. [**Table 18 on page 117**](page 117) lists the LRFs that are in the `\usr\ov\lrf` directory.
Table 18. Local Registration Files

<table>
<thead>
<tr>
<th>lrfile</th>
<th>daemon name</th>
</tr>
</thead>
<tbody>
<tr>
<td>netmon.lrf</td>
<td>netmon</td>
</tr>
<tr>
<td>ovesmd.lrf</td>
<td>ovesmd</td>
</tr>
<tr>
<td>ovelmd.lrf</td>
<td>ovelmd</td>
</tr>
<tr>
<td>ovtopmd.lrf</td>
<td>ovtopmd</td>
</tr>
<tr>
<td>ovwdb.lrf</td>
<td>ovwdb</td>
</tr>
<tr>
<td>pmd.lrf</td>
<td>pmd</td>
</tr>
<tr>
<td>snmpCol.lrf</td>
<td>snmpCollect</td>
</tr>
<tr>
<td>trapd.lrf</td>
<td>trapd</td>
</tr>
</tbody>
</table>

**Note:** The `\usr\ov\conf\ovsuf` file contains the path name to each object manager. The object manager executable can be removed or moved from its expected location. By default, the managers are in `\usr\ov\bin`.

If some lrfile files are missing, find the files in the `\usr\ov\newconfig` directory by using this command:

```
dir \usr\ov\newconfig
```

Search the directory listing for the missing lrfile files. Copy the files into the `\usr\ov\lrfile` directory. Then issue the `ovaddobj` command:

```
oveaddobj [-t <target_hostname>] <lrfile_filename> [remote_hostname]
```

The `hostname` variables specify a system name that must be defined in the hosts database used by the system. If either `hostname` is not specified, it is assumed to be the system on which the `oveaddobj` command is being run.

The `lrfile_filename` variable specifies an lrfile file that must contain information about a single object manager and the objects it manages. For more information about lrfile files, refer to the *Tivoli NetView for Windows Programmer’s Guide*.

**Stopping the NetView Program**

To stop the NetView program, select *File..Exit* from the NetView main menu. Although this stops the NetView console, the NetView daemons continue running and collecting topology information. After the NetView program has stopped, you can stop the daemons by double-clicking on the *Stop Daemons* icon in the NetView program group.

If you want the NetView program to continuously monitor your network, do not take any of the following actions. Any of these actions will stop the NetView daemons that monitor network status and collect topology information:

- Select the Stop Daemons icon
- Select Shutdown
- Turn off the system

The NetView program runs the NetView daemons as a service. The NetView service provides the ability to run the NetView daemons automatically on system startup and to continue running daemons across login sessions.

**Note:** Locking your PC or using a screen saver does not stop the daemons.
Diagnosing Network Problems

Many network variables affect the problem determination process for the NetView program including:

- The size of the network
- The type of systems on the network
- Where the systems reside
- The presence of bridges on the network
- The presence of gateways and routers on the network

The best approach to diagnosing network problems is to begin with the lowest level (physical connections, data links, network protocol, transport) and work up to the highest level (session, presentation, and application). You can rule out problems within the local network and begin isolating problems across networks. You might need to refer to the communications documentation for your specific hardware or operating system.

Before you start to diagnose network problems, allow netmon sufficient time to discover your network. The amount of time required by netmon to discover your network depends on how many nodes you have on your network, what type of system you have, and whether or not the nodes support SNMP. The general guideline is that the average node discovery time is about five seconds per node, but some nodes might take up to two minutes.

The netmon daemon discovers the nodes on your IP network by using SNMP requests to gather node information: Make sure the netmon daemon has access to this information.

- The subnet mask from the agent on the management system. To get the subnet mask:
  1. Double-click on the Control Panel icon in the Windows Main program group.
  2. Double-click on the Network icon in the Control Panel.
  3. Select TCP/IP Protocol from the list of installed software.
  4. Click on the Configure button.
- The name of the default router in the management system’s routing. You can enter the NetView command \usr\ov\bin\rnetstat -r to check this.
- SNMP information from the SNMP agent software on the management system.

For the netmon daemon to work, these conditions must exist:

- The management system must be correctly configured for networking.
- The management system must be running an SNMP agent.
- Nodes must be up and responding to ping requests to be discovered.

The netmon daemon is the most efficient in these situations:

- The more routers running SNMP agents in the network, the better.
- The more nodes running SNMP, the better.
- The more SNMP information available, the better (for example, from valid ARP Caches).

Information about the discovered nodes is stored in the topology database and is used to automatically generate the network map. The topology database is in the \usr\ov\databases\openview\topo directory. To view the contents of the database, run the ovtopodump utility from a command prompt window.

Over time, the netmon daemon discovers all new nodes on the network that are one hop away from the management system. One hop includes all the nodes in a subnet, up to the management system’s side of the router.
However, if a new node never communicates with a gateway or other nodes on the network that support SNMP, the manager might not find it. In these instances, you can add the node manually by selecting Object..New.

You can also manage a subnet by selecting it and then selecting Object..Manage on the NetView main menu. Additionally, you can use seed files to force discovery past one hop. Select Options..Server Setup from the NetView main menu. Then select the Daemons tab page and select Netmon Daemon from the drop-down list box. Edit the Seed File for Discovery field in the dialog box that is displayed.

The event browser shows each node as it is discovered. This enables you to monitor the discovery process. For more information, see "IP Discovery and Layout Problems".

---

**IP Discovery and Layout Problems**

IP discovery and layout problems can occur for these reasons:
- IP discovery and layout might not discover all the nodes on your network.
- The netmon daemon might not discover any nodes on your network, or your map might be drawn incorrectly.

**Why IP Discovery Might Not Discover All Nodes**

The netmon discovery process does not discover all the nodes on the network if these conditions exist:
- The network management system’s default router does not support SNMP, or the network has few SNMP nodes.
- The nodes on the network do not talk to each other.
- The nodes only talk among themselves in an isolated area that is connected to the network through a gateway that does not support SNMP.
- The network uses bridges and not routers. Bridges do not supply any information about what nodes are on the network.
- The management system subnet mask is set wrong.

**What to Do if IP Discovery Fails**

Follow the items in this checklist if the netmon daemon fails to discover network nodes:
- Verify that the Windows SNMP Service is started.
  1. Double-click on the Control Panel in the Windows Main program group.
  2. Double-click on the Services icon.
  3. Check SNMP Service. If SNMP Service is not started, start it.
- Ensure that the nodes are running and that the nodes are responding to pings and SNMP requests. Issue the ping command and the SNMP request `snmpwalk nodename system` from a command prompt window, or double-click the MIB Browser icon in the NetView program group.
  If there is no SNMP response, the node is not running SNMP, or the node is using a community name that does not allow SNMP access. Find out what community name is in use and configure NetView to use that name whenever accessing that node. This can be done by running the snmpconf.exe utility.
- Ensure that the routing tables are configured properly. Enter the `\usr\ov\bin\rnetstat -r` command in a command prompt window. The netmon daemon uses the default router in the routing table to start the discovery process.
See "Using a Seed File to Control Network Discovery" on page 89 for information on using a seed file to accelerate initial topology map generation.

- Ensure that the information the IP Map application has configured for the map object is accurate. To verify this information, select Edit..Properties from the menu bar, then select IP Map from Applications and click the Properties to see the IP Map properties.

- Ensure the subnet mask is correct. The subnet mask on the management system is used for the entire network. If the subnet mask on the management system is incorrect, discovery fails. Symptoms of incorrect subnet masks include:
  - The netmon daemon might discover nodes from the Internet that you do not know about.
  - Nodes are displayed in a different subnet than you expected.

To determine which subnet mask the SNMP agent returns to the netmon daemon, do one of the following:

- In the submap window, select the node running the NetView program, and then select Object..Object Properties. Select the Other tab and then select IP Map. The IP Map object properties are displayed.

- In the submap window, select the node running the NetView program and query the following MIB object by selecting Tools..MIB..Browser:
  `.iso.org.dod.internet.mgmt.mib2.ip.ipAddrTable`  
  `.ipAddrEntry.ipAdEntNetMask`

- Execute the snmpget or snmpwalk command from a command prompt window.

Resolving an Incorrect Subnet Mask

To quickly identify all the nodes in your network with incorrect subnet masks, go to the "Bad Mask" SmartSet submap. You can also activate a log of these nodes from the netmon daemon page in the server setup. You can use a similar procedure to find nodes with the SNMP System Object ID not recognized by NetView. An unrecognized SNMP System Object ID (OID) can be corrected by adding the appropriate entries to the `\usr\ov\conf\oid_to_type` and `\usr\ov\conf\oid_to_sym` files.

To solve map problems caused by an incorrect subnet mask, follow these steps:

1. Correct the problem with the subnet mask by selecting Control Panel..Network from the Windows Main program group. Select TCP/IP Protocol and click the Configure button. Change the subnet mask to reflect your network’s addressing scheme. Restart your system.

2. Clear out the databases and repopulate them by clicking Start..Programs..NetView..Administration..Server Setup and then selecting Clear and Re-create All NetView Databases from the Databases tab page.

The NetView program supports multiple subnet masks on the network. The term network is defined similarly as it is defined by TCP/IP. There are three primary classes of networks: A, B, and C. Network addresses consist of four octets separated by a decimal point:

- Class A networks are identified using the first octet, for example, 15. Or, if the first bit of the network address is zero, this also signifies a Class A network.

- Class B networks are identified by the first two octets, for example, 128.1. Or if the first two bits of the network address are 10, this also signifies a Class B network.
• Class C networks are identified by the first three octets, for example, 192.0.2. Or if the first three bits of the network address are 110, this also signifies a Class C network.

Performance During Autodiscovery

Initial autodiscovery is resource intensive. The NetView status bar contains Network Monitor Activity bars that enable you to monitor the progress of autodiscovery. Look for the three bars in the status to the right of the node count. The top bar displays new node polling activity, the middle bar displays node configuration polling activity, and the bottom bar displays status polling activity.

In general, more CPU cycles are consumed for SNMP operations than for ICMP (ping) operations. A high number of SNMP operations being performed at a given instant can affect the performance of your system. If the performance with the default values is sluggish, you can adjust the overall autodiscovery workload by changing the Discovery Speed under: Options -> Server Setup -> Discovery.

If all nodes in your network are connected by interfaces that have explicit IP addresses, you can uncheck the field Discover IP Nodes beyond Unnumbered Interfaces under: Options -> Server Setup -> netmon Daemon. This decreases network traffic.

ICMP operations are usually not a significant contributor to performance degradation. An exception is if the use of Ping Spray is enabled and the networks being discovered are sparsely populated; in sparse networks there can be potentially many ICMP timeouts and retries that occur due to ICMP messages being sent to non-existent IP addresses. You can change the use of Ping Spray for your site and adjust the Ping timeout values using: Options -> Discovery.

See “Configure Autodiscovery for Your Site” in the online help file.

Clearing and Repopulating NetView Databases

If a map that you expect to see is not displayed by the NetView program, you may need to clear the NetView databases and repopulate them. Possible reasons for clearing and repopulating the databases include:
• Problems discovering nodes
• Damage to the network database
• Database not valid because of many changes in the network
• The NetView program was moved to another system

You can clear the NetView databases and repopulate them by selecting Options..Server Setup, the Databases tab page, then selecting Clear Databases from the drop-down list.

Unexpected Networks Are Displayed

If unexpected networks appear on the submap, such as a network having a Class A, Class B, or Class C network number when you expect a subnet network number, check to ensure that the nodes appearing on those networks have correctly configured subnet masks. Incorrect subnet masks can cause incorrect networks to be added to the submap. The netmon daemon might discover and add nodes from outside your administrative domain to networks without a subnet or an incorrect subnet network.
The NetView program can only draw networks correctly if only one subnet mask per network class is used. Also, all IP interfaces on the following objects must have been correctly configured:
- Gateways and routers
- Management system
- Nodes in a seed file for netmon

**Inaccurate Map Object and Database Information**

If isolated information about the NetView program's map objects appears to be inaccurate, follow these steps:

1. Select **Edit..Properties** and then select **Map** tab page to correct the inaccuracy.
2. Select **Test..Demand Poll** to update the database entries for the nodes or verify monitoring of the nodes.
3. Dump and verify the database by selecting **Options..Server Setup** from the NetView main menu. In the NetView Server Setup dialog box, click on the **Database** tab, then select **Verify and Dump Database** from the drop-down list. You can also use the `ovmapdump` command to view the contents of the database for the objects in question.
4. If a node that is a gateway appears as a host, ensure that the node supports SNMP and multiple interfaces, and that the MIB oid_to_type variable `ipForwarding` is set to **TRUE** (value of 1).
   Select **Object...New** to add a gateway, then select **Object...Delete** to delete the host object.

If the majority of the topology database appears to be inaccurate, perform these actions:

1. Shorten polling intervals for better monitoring accuracy by selecting **Options..Polling**. Be aware that this generates more polling traffic on the network.
2. Select **Options..Server Setup** from the menu bar.
3. Select the **Databases** tab page.
4. Select **Verify and Dump Database** from the selection list.
5. Click **OK**.

You can also use the `ovmapdump` command to dump part or all of the topology database for inspection. The listing might be large.

**Map Discovery Problems**

If the netmon daemon does not initially locate (discover) nodes, check these items:

- Ensure that the SNMP Services are installed and running on the management system. The NetView program requires that the SNMP Services are operating and configured correctly with the right community information.
- Check the network mask:
  1. Double-click **Control Panel**.
  2. Double-click **Network**.
  3. Select **TCP/IP Protocol** from the list of installed network software.
  4. Click **Configure**.
- Check the `<winnt>\system32\drivers\etc\hosts` file to verify the management system address.
• Ensure that the SNMP agents are installed and running on the agent systems, particularly on gateways. The problem might be that there are no SNMP agents installed.

If you specified a seed file, ensure that the seed file exists and you are using the correct name. The problem might be that the specified seed file on the netmon command line in the \usr\ov\conf\ovsuf file does not exist. Also, make sure that you use all the nodes that you want to use as seeds in the seed file and that all the nodes listed in the seed file support SNMP. Bridges and hubs cannot be used as seeds.

• Ensure that the directory \usr\ov\databases\openview exists and that none of the files are read-only.

• Check if your routers or gateways support secondary addressing. If the devices do support secondary addressing, and the secondary addresses were not found during discovery, select Options..Server Setup..Daemons..Netmon to have netmon check for these addresses.

• Ensure you can ping the default router and the closest SNMP supporting router.

• Ensure the default route is configured correctly. To find the default route, use the \usr\ov\bin\rnetstat -r command.

Over time, the NetView program discovers all nodes up to one hop from the management system that communicate with IP. The time it takes varies, depending on machine size and speed, paging space, network size, and other variables.

One hop includes all the nodes in a subnet, up to the management system's side of the router. You must manage a subnet or use seed files to force discovery past one hop.

However, if a new node never communicates with a gateway or other nodes on the network that support SNMP, the NetView program might not find it. Some nodes, such as bridges, have IP addresses, but do not usually communicate with IP stations, although they might forward IP packets. These nodes are not discovered unless there is some IP communication to them.

**Routers Are Not Discovered**

Check to see if the routers are backup routers, that is, routers used in an emergency when another router goes down. The netmon daemon might not discover backup routers, because it looks only for nodes on the network that are actually used. Therefore, a backup router is discovered either as a host or it might not be discovered at all. To solve this problem, follow these steps:

1. Manually, add the router to the submap. For information about how to manually add the router, see "Add Objects" in the online help.
2. Ensure that the SNMP agent is running on the router.

**Objects Are Not Displayed**

If nodes are not displayed on the map after starting the NetView program, ensure the following daemons are running: ovspmd, ovwdb, trapd, ovtopmd, and netmon.

If any of the previous daemons are not running, exit the NetView program, stop the daemons, and restart the NetView program.

If you have installed the NetView program on an Alpha system with less than 64 MB of memory, which is less than the minimum configuration, objects may not be displayed on the map. Upgrade your system to a minimum of 64 MB.
Objects Are Placed in Incorrect Segments
When a node is discovered and responds to Pings, but does not have an SNMP agent that responds to queries, the NetView program may not be able to gather sufficient information about the interface of the node to determine the default segment type. If the subnet in which the node is contained does not yet have a segment type, the netmon daemon attempts to create the default segment type.

If you have set the default segment type, but this segment type is not the type that is created, it may be due to the following reason. The account that you are logged into is not the NetView account, and the default segment type was set while using the non-NetView account, log out of the current account and log into the NetView account. Then, set the default segment type using: Options -> Server Setup -> Daemons -> Netmon Daemon.

Unnumbered Serial Links on Cisco Routers Are Not Discovered
- Ensure the IP Nodes beyond Unnumbered Interfaces option is checked on in the netmon daemon page of Server Setup by selecting Options -> Server Setup. (This corresponds to the -u switch in netmon.lrf and has always existed in previous versions as the default).
- Ensure that both ends of the unnumbered serial link are discovered and managed. If necessary, use a seed entry in the netmon seed file to make sure that both routers are discovered.
- Demand poll both ends of the unnumbered serial link. If only one end is discovered, demand polling it should trigger the other end to be discovered.
- If you are still having problems discovering routers, stop netmon and run the command:\usr\ov\bin\ovtopofix -a.
- If both routers have been discovered and the link is still not displayed after demand polling both routers, check the Cisco CdpCache SNMP next hop entries by typing:
\usr\ov\bin\snmpwalk routernamex.1.3.6.1.4.1.9.9.23.1.2.1.1.6

Identify the entry for the next hop in question and verify using nslookup that the IP name can be resolved on the NetView machine correctly. Multiple IP addresses are valid, but netmon may resolve the name to any one of these addresses. Problems can arise if the domain name of the IP name is missing. In that case, you can add this IP domain to the Domain Suffix Search Order list on the DNS page of the TCI/IP Properties dialog, which is available from the Network applet on the Control Panel or add an entry to the %SystemRoot%\system32\drivers\etc\hosts file.

If one of the routers has an HSRP virtual interface active, ensure that netmon is configured for secondary addresses. This is the default and corresponds to the -S switch in netmon.lrf. Otherwise, if the next hop resolves to the HSRP virtual address and it is not being discovered, the link may not get created either.

Nodes Are Not Discovered Beyond the Router
Determine whether the router has a serial link or an X.25 link. When netmon pings a serial link or an X.25 link, the ping command might time out. In this case, the netmon daemon cannot find those nodes. To solve this problem, use one of these methods:
- Add the routers on the other side of the serial links and the X.25 links to the seed file and restart netmon.
- Manually, add the routers to the map and select Options..Polling to configure a longer time out for those particular networks.
SNMP Nodes Are Incorrectly Identified

The problem might be that all the netmon SNMP requests to a node fail to get a system description. During SNMP requests, the netmon daemon asks for system, interface table, IP address table, and other information in a specific order. The netmon daemon queries as follows:

- Configuration checking at initial discovery
- Demand Polling
- Automatic daily configuration checking

If one or more queries succeed, the netmon daemon assumes the node supports SNMP. If Demand Poll fails (due to a busy transport), wait about two minutes and try again.

The most common reason for netmon failing to report an SNMP node is that the node community name is not public and is not entered or incorrectly entered in the `\usr\ov\conf\ovsnmp.conf` file. If the community name is not configured or if the manager uses the wrong community name in the request, the SNMP agent does not return a message indicating a wrong community name.

To change community names, select Options..SNMP from the main menu, select a node in the table, and then click on the Properties button.

Another common reason for Demand Poll failure is that the agent returns incorrect information. To verify a problem with the SNMP agent, select Tools..MIB..Browser. You can also issue the `snmpwalk` command on the tables shown in Table 19, for example,

```
snmpwalk -c public <node_name> ip
```

**Table 19. Problem Verification Option**

<table>
<thead>
<tr>
<th>Table</th>
<th>OID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface table</td>
<td>.iso.org.dod.internet.mgmt.mib-2.interfaces</td>
</tr>
<tr>
<td>IP address table</td>
<td>.iso.org.dod.internet.mgmt.mib-2.ip.ipAddrTable</td>
</tr>
<tr>
<td>ARP Cache table</td>
<td>.iso.org.dod.internet.mgmt.mib-2.ip.ipNetToMediaTable</td>
</tr>
</tbody>
</table>

To diagnose the problem further, answer the following questions:

- Is the information that is being returned readable?
- Do the entries in the interface table have a corresponding entry in the IP address table?

If you have verified that the problem is with the SNMP agent, send the information to the vendor of the agent.

A Router Is Displayed as a Host System on the Map

Verify these items:

- Ensure the community name on the agent and the manager match.
- Check that IP forwarding is set to forwarding and that the router has more than one interface card. You can select Tools..MIB..Browser to see what values are returned by the particular routes.
- Check the oid_to_sym object identifier.
Diagnosing Problems with Topology, Discovery, and Database Daemons

The NetView program discovers and updates the topology of an IP network and translates that information into symbols on its map. The NetView daemons that perform these tasks can be divided into three categories: topology, discovery, and database daemons.

The netmon, trapd, pmd, ovwdb, and ovtopmd daemons should be running when the NetView program is running. You can double-click on the Daemon Status icon in the NetView program group (or issue the ovstatus command) to see if these daemons are running.

The topology, discovery, and database daemons are shown in Table 20.

Table 20. Topology, Discovery, and Database Daemons

<table>
<thead>
<tr>
<th>Daemon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ovtopmd</td>
<td>Maintains the network topology database</td>
</tr>
<tr>
<td>ovwdb</td>
<td>Controls the OVW object database</td>
</tr>
<tr>
<td>netmon</td>
<td>Discovers and monitors nodes on an IP network by polling SNMP agents</td>
</tr>
<tr>
<td>ovspmd</td>
<td>Manages the NetView daemons processes, including stopping, starting and reporting on their status.</td>
</tr>
</tbody>
</table>

ovtopmd Daemon

The ovtopmd daemon maintains the IP topology database used by the netmon and ipmap daemons. It enforces rules for IP topology accuracy and caches the data to maximize system performance. When you click on the Daemon Status icon, you may receive an error message similar to this:

`ovspmd: \usr\ov\conf\ovsuf: object manager ovtopmd not registered`

Check whether the `\usr\ov\conf\ovsuf` file is damaged.

If the Daemon Status report indicates the ovtopmd daemon is not running, enter:

`ovstart -v`

The `-v` (verbose) option creates a report for each object manager, whether it is started or not. Generally, this starts the ovtopmd daemon and all daemons on which the ovtopmd daemon depends.

The ovtopmd daemon depends on the ovwdb and trapd processes being started. If the ovwdb and trapd daemons do not start, ovtopmd does not execute successfully. In addition to the ovwdb daemon, the ovtopmd daemon relies on the field definitions being loaded into the ovwdb database, which generally happens during NetView installation or when the `ovw_fields` command is issued.

If the ovtopmd daemon gives an error indicating that it was unable to map a field name into a field ID, follow these steps:

1. Verify that LANG is set to `c` and that the `\usr\ov\fields\c` directory exists and contains at least the following files: ovw_fields, ip_fields, and snmp_fields. The LANG environment variable provides national language support. You can view the environment variables by selecting Control Panel..System in the Windows Main program group.
2. Load necessary field definitions into the ovwdb database using the \texttt{ovstart}, \texttt{ovwdb}, and \texttt{ovw_fields} commands.

3. Start the ovtopmd and netmon daemons.

\textbf{Checklist for ovtopmd Daemon Problems}

By default, the ovtopmd daemon uses a TCP socket bound to port 8880. If the daemon indicates an error attempting to bind to this port number, because another application is already using it, use this checklist:

- Ensure that other instances of the ovtopmd daemon are not using port 8880.

  The most common reason for errors binding to the TCP socket is the presence of another ovtopmd daemon that was started by a means other than the \texttt{ovstart} command. In this case, both instances of the daemon must be stopped by using the Process Viewer to kill the ovtopmd process. Daemons started by the \texttt{ovstart} command can be stopped by using the \texttt{ovstop} command.

- Change the port number by changing the ovtopmd entry in \texttt{\windir\system32\drivers\etc\services}. Choose a number other than 8880 that does not conflict with other services listed in \texttt{\etc\services}.

\begin{verbatim}
  ovtopmd 8880\tcp # NetView IP Topology daemon
\end{verbatim}

- Stop and then restart the daemons.

- Start the NetView program by double-clicking on the \texttt{NetView} icon.

- If the ovtopmd daemon still fails to run successfully, contact your NetView support representative.

When the ovtopmd daemon has been started, you can verify that it is responding to requests by executing the \texttt{\usr\ov\bin\ovtopodump} command. By default, the \texttt{ovtopodump} command prints a single line of output for each object in the topology. To get longer listings that provide more information about each object enter \texttt{ovtopodump -rl}. To get more information about a single object, enter \texttt{ovtopodump -l <object_name>}.

If you enter \texttt{ovtopodump -l}, the result looks similar to this:

\begin{verbatim}
  TOP0 OBJECT ID: 70
  TIME CREATED: 05/12/95 08:39:17
  TIME MODIFIED: 05/12/95 01:31:10
  GLOBAL FLAGS:
  NUMBER OF NETWORKS: 13
  NUMBER OF SEGMENTS: 19
  NUMBER OF NODES: 238
  NUMBER OF INTERFACES: 281
  NUMBER OF GATEWAYS: 21
\end{verbatim}

For more information about the \texttt{ovtopodump} command, see the \textit{Tivoli NetView for Windows Programmer's Reference}.

The \texttt{ovtopodump} command prints summary information for the entire IP topology.

If the command fails, this indicates some problem with the ovtopmd daemon. Try stopping all daemons with the \texttt{ovstop} command, then restart them with the \texttt{ovstart} command. If the \texttt{ovstart} command succeeds, but the ovtopmd daemon still fails, verify that at least one of the following is true:

- The IP address of the loopback interface is 127.0.0.1. Check this using the \texttt{snmpwalk <nodes> IP.IPAddrTable.ipAddrEntry} command.

- There is an entry for localhost that has the correct address for the loopback interface in the \texttt{\windir\system32\drivers\etc\hosts} file.
• The IP address for the local host name is correct. Ping the node and verify the address.

You can run the **ovtopofix** command to detect and correct inconsistencies between the IP topology database maintained by ovtopmd and the database maintained by ovwdb. The default behavior is to remove old hints from the ovwdb database, and to verify the managed and removed state of all objects. When used before updates to the databases are attempted, the **ovtopofix** command makes sure that both netmon and ipmap are not running. For more information about the **ovtopofix** command, refer to the *Tivoli NetView for Windows Programmer’s Reference*.

**ovwdb Daemon**

The ovwdb daemon controls the OVW object database. The ovwdb daemon should be running when the NetView program is running. Normally, it is started through the **ovstart** command. To verify whether it is running, double-click on the **Daemon Status** icon (or issue the **ovstatus ovwdb** command).

You should see information similar to this example:

```plaintext
object manager name: ovwdb
behavior: OVs_WELL_BEHAVED
state: RUNNING
PID: 4115
last message: Initialization complete.
exit status: -
```

**Checklist for ovwdb Daemon Problems**

By default, the ovwdb daemon uses a TCP socket bound to a port of 8879. The port is located in the services file, `<winnt>\system32\drivers\etc\services`. If an error is indicated when you attempt to bind to this port number because another application is already using it, follow these steps:

1. Ensure that another ovwdb daemon is not using port 8879.
   
   The most common reason for errors binding to the TCP socket is the presence of another ovwdb daemon. If an ovwdb daemon was started without the **ovstart** command, the **ovstop**, **ovstart**, and **ovstatus** commands do not recognize this daemon. To prevent this problem, always start all NetView daemons with the **ovstart** command and stop them with the **ovstop** command.
   
   If an application is also using port 8879, continue with the next step.

2. Change the port number by changing the entry for the ovwdb daemon in the `<winnt>\system32\etc\services` file as shown in this example:

   ```plaintext
   ovwdb       8879\tcp   # NetView Object Database daemon
   ```

3. If the ovwdb daemon still fails to run successfully from the **ovstart** command, contact your NetView support representative.

**netmon Daemon**

The netmon daemon uses various ICMP Echo requests (ping) to poll IP and non-SNMP nodes for status information, and SNMP requests to poll SNMP nodes for MIB values. When the netmon daemon detects a change, it updates the topology database through ovtopmd with the change and sends the appropriate event to the trapd daemon. The trapd daemon logs the event and forwards it to the ipmap application and events application. The event is also forwarded to the NetView program.
The ipmap application updates the topology map, and the events application updates the Monitor..Events displays accordingly. See Figure 30 for an illustration of these netmon daemon interactions.

You can control netmon polling by selecting Options..Polling and Options..SNMP from the NetView main menu. Options..Polling stores the netmon polling intervals in the \usr\ov\databases\openview\topo\polling file and Options..SNMP stores SNMP polling information for netmon in the \usr\ov\conf\ovsnmp.conf file.

Note: Only modify these files by selecting Options..Polling and Options..SNMP as they contain order-sensitive data.

The nmpolling application checks periodically to see if the netmon, ovtopmd, and snmpCollect daemons are running. If one of the daemons is not running, the nmpolling application displays a dialog box and reports the inactive process. For information on changing the polling interval or on disabling this polling feature, refer to “Setting and Changing Polling Intervals” on page 93 and nmpolling in the Tivoli NetView for Windows Programmer’s Reference.

The netmon daemon polling is scheduled on a per-node basis, but polling a node might not occur when it is scheduled due to the polling load for a particular time.
interval. Also, polling of manually added nodes and nodes changed from an unmanaged state to a managed state might not occur immediately after they are added or changed to a managed state.

Only one instance of the netmon daemon is permitted to run on the management system.

**Checklist for netmon Daemon Problems**

Use the following checklist to diagnose netmon daemon problems:

- If the netmon daemon stops because there is no free space, the database is damaged.
- Select Test..Demand Poll to test netmon polling activity to a node. This test causes netmon to immediately poll the selected node.
- Check the log file. The default log file is \usr\ov\log\netmon.trace. Or, you can select Monitor..Events..Selected with a filter for netmon errors to see events for a specific node.
- Ensure that adequate paging space is available. At least 120 MB is recommended.
  1. Double-click Control Panel.
  2. Double-click the System icon.
  3. Click Virtual Memory to check on available paging space. If there is not enough, increase the paging space.
- Use the netmon daemon tracing to help isolate problems. Modify the file \usr\ov\conf\ovsuf to add -t <tracefile> and -M <tracefile_mask> to the netmon entry. This causes netmon to log the results of its ICMP and SNMP requests. For more information, see “Using Daemon Tracing” on page 133.

**Solving IP Name/IP Address Mismatch Problems**

If the IP name and IP address of the management station, as configured for TCP/IP and stored in the registry, are different from the values stored in the local hosts file <SystemRoot>\system32\drivers\etc\hosts, the NetView program does not function properly.

To avoid this situation the NetView installation procedure updates the <SystemRoot>\system32\drivers\etc\hosts file with the IP name and IP address of the management station from the TCP parameters stored in the registry, if there is no entry present. Also, the NetView ovwdb daemon checks whether the host name and address stored in the registry are consistent with those stored in the hosts file. If the two views do not match, the following warning message is displayed:

An IP Address mismatch between the hosts file and the registry has been detected.

This should be corrected as soon as possible; however, ovwdb continues to complete the installation.

On startup, the netmon daemon also checks that the two views match. If they do not match, the following error message is displayed and netmon exits:

An IP Address mismatch between the hosts file and the registry has been detected:

hosts file:
registry:
You must correct the hosts file and restart the netmon daemon before discovery can occur.

If you want to prevent netmon from exiting, set the environment variables as follows:

```
NV_NOIPCHECK =1
```

However, error messages are still written to the `\usr\ov\log\nv.log` file and the problem persists.

**ovspmd Daemon**

The ovspmd daemon starts and manages the NetView daemons. If the ovstatus command shows that any of the daemons are running, then the ovspmd daemon is running.

The ovspmd daemon is automatically started when you start the daemons. If you determine that a problem is related to the ovspmd daemon and not another daemon, check to see if another daemon is not running.

**A Daemon Is Not Running**

If the output from the ovstatus command indicates that a particular daemon is not running, use the ovstart ovspmd command to start ovspmd. This command should start the daemons that are not running.

If an error message shows that a daemon failed to start, follow these steps:

1. Use the ovstatus ovspmd command to show the status of the daemon.
2. If a daemon is running under the ovspmd daemon, use the ovstop command to stop it.
3. Check that the daemon has stopped.
4. If the daemon did not stop, double-click on the Process Viewer icon in the NetView program group, then kill the daemon.
5. Execute the ovstart -v command to restart the daemon and to verify that it starts properly.

**The ovspmd Daemon Is Not Running**

If you get the following error messages:

```
ovstart: ovspmd is not running. It might have failed to start.
ovstart: unable to contact ovspmd: error message
```

1. Ensure the `\<winnt>\system32\drivers\etc\services` file has the following entries:

   - OVSPMD_REQ 8883
   - OVSPMD_MGMT 8884

   If the entries are not there, add them to the services file.
2. Stop all NetView processes that are running by using the ovstop command and restart the processes with the ovstart command.
3. If ovstart still fails, contact your NetView support representative.

---

**Diagnosing Problems with Event and Trap Processing Daemons**

An event is an occurrence of significance to a task, such as an SNMP trap or a NetView internal event. A trap is an unsolicited event generated by an agent and forwarded to a manager. Traps inform the manager of changes that occur in the network. The daemons shown in Table 21 on page 132 monitor event and trap
Table 21. Monitor and Trap Daemons

<table>
<thead>
<tr>
<th>Daemon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>trapd</td>
<td>Receives events and traps from SNMP agents and internal daemons, logs them, and forwards them to the appropriate daemon or application.</td>
</tr>
<tr>
<td>snmpCollect</td>
<td>Collects, compares, and stores SNMP MIB values</td>
</tr>
</tbody>
</table>

trapd Daemon

The trapd daemon receives events and traps from SNMP agents and internal daemons, stores them in a SQL database, and forwards them to registered applications, including ipmap, netmon, snmpCollect, nvcord, collmap, and nvcold. Events can be generated as shown in Table 22.

Table 22. Processes that Generate Events

<table>
<thead>
<tr>
<th>Process</th>
<th>Generates events when:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipmap</td>
<td>You edit the map or when the status of a node propagates to segments and networks.</td>
</tr>
<tr>
<td>snmpd and other agents</td>
<td>An SNMP trap is sent.</td>
</tr>
<tr>
<td>netmon</td>
<td>A change is detected as a result of polling nodes or as a result of receiving a trap forwarded by the trapd daemon.</td>
</tr>
<tr>
<td>ovtopmd</td>
<td>Topology changes occur.</td>
</tr>
<tr>
<td>snmpCollect</td>
<td>A threshold is exceeded or reset.</td>
</tr>
<tr>
<td>trap, loadmib, collect</td>
<td>You edit configuration files.</td>
</tr>
</tbody>
</table>

When the netview.exe application generates an event:

1. The IP Map application sends the event to the trapd daemon and updates the topology database with the map change responsible for the event.
2. The trapd daemon stores the event and forwards it to netmon and the events browser.

Checklist for trapd Daemon Problems

If you suspect a problem with the trapd daemon, try the following actions to further isolate the problem:

- Look at the nv.log file for trapd process errors. The default log file is /usr/ov/log/tnv.log.
- Type event at the command prompt and check that a "Node Up" trap appears in the Event Browser or SQL database.
- If the trapd daemon appears to be hung, stop all the daemons by double-clicking on the Stop Daemons icon in the NetView program group or by using the ovstop command.
- Restart all the daemons by double-clicking on the Start Daemons icon or by using the ovstart command.
- Ensure that adequate paging space is available:
  1. Double-click the Control Panel icon.
  2. Double-click the System icon.
  3. Click Virtual Memory to check on available paging space. The recommended minimum is 120 MB. If necessary, increase the paging space.
snmpCollect Daemon

If the snmpCollect daemon is not collecting data properly, check these conditions:

- The collection mode for the collection source is not store.
- Collections are suspended for this MIB object.
- The current configuration has not been applied through the Apply or OK buttons.
- The specified data cannot be collected, possibly due to a MIB instance that is not valid. Look in \usr\ov\log\snmpCol.trace for errors or use the snmpCollect -S command to show the configuration.
- The snmpCollect daemon has not had time to collect the data. This can take twice as long as the collection Polling Interval after the message Beginning data collection has been logged to \usr\ov\log\snmpCol.trace.
- The node is unmanaged.
- The snmpCollect daemon is not running.

Dumping Data from snmpCollect Daemon

To dump the data collected by the snmpCollect daemon, use the snmpodump command. The output contains the date and time the data was collected, the domain of the node on which the data was collected, and the value of the collected MIB variable. For more information about the snmpodump command, refer to the Tivoli NetView for Windows Programmer’s Reference.

Checklist for snmpCollect Problems

Check the following items to diagnose the SNMP subsystem:

- Verify that community names are correctly configured on both the management and agent systems.
- Verify that an SNMP agent process is running on the agent system.
- Select Test..Connectivity to verify SNMP operation on a remote SNMP node.
- Check the agent log file, \usr\ov\log\snmpCol.trace, on the agent system for errors.

Checklist for Agent MIB Problems

Use the following checklist to diagnose agent MIB problems:

- Select Tools..MIB..Browser to verify that the information retrieved is accurate.
- Use the snmpget command from the management system to inspect an individual agent MIB value.
- Use the snmpwalk command from the management system to dump part or all of the agent’s MIB for inspection.
- Explore the system and network configuration on the agent system to verify configuration accuracy.

Using Daemon Tracing

Daemon tracing is the recording of normal and abnormal network activities. It helps reconstruct a sequence of events, both normal and abnormal. Daemon tracing can generate large trace files. To maintain system performance, perform tracing and logging only when necessary, and monitor the size of trace and log files frequently. Before starting a daemon for tracing, be sure that the daemon has been stopped.

Use Table 23 as a reference when you trace NetView daemons. The tracing commands assume the daemon is already running and send a message to the currently executing daemon to update its trace mask. Issue the same command a second time to turn off tracing. Daemon options and command formats are listed in

Chapter 10. Diagnosing and Solving Problems 133
Table 23. Daemon Tracing

<table>
<thead>
<tr>
<th>Daemon</th>
<th>Tracing Default</th>
<th>Default Log File</th>
<th>Start/Stop Tracing</th>
</tr>
</thead>
<tbody>
<tr>
<td>\usr\ov\bin\netmon</td>
<td>Off (0)</td>
<td>\usr\ov\log\netmon.trace</td>
<td>Use netmon -M tracemask to start tracing or netmon -M 0 to stop</td>
</tr>
<tr>
<td>\usr\ov\bin\snmpCollect</td>
<td>Off</td>
<td>\usr\ov\log\snmpCol.trace</td>
<td>Use snmpCollect -T</td>
</tr>
<tr>
<td>\usr\ov\bin\trapd</td>
<td>Off</td>
<td>\usr\ov\log\trapd.trace</td>
<td>Use trapd -T</td>
</tr>
</tbody>
</table>

A **tracemask** is a value that specifies one or more types of data to be sent to a trace file. The tracemask values are listed in Table 24.

Table 24. Trace Mask Values for the netmon Daemon

<table>
<thead>
<tr>
<th>Value</th>
<th>netmon</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Turn off tracing</td>
</tr>
<tr>
<td>1</td>
<td>Trace ICMP echo requests</td>
</tr>
<tr>
<td>2</td>
<td>Trace ICMP echo replies and timeouts</td>
</tr>
<tr>
<td>4</td>
<td>Trace SNMP requests</td>
</tr>
<tr>
<td>8</td>
<td>Trace SNMP replies and timeouts</td>
</tr>
<tr>
<td>16</td>
<td>Trace traps generated</td>
</tr>
<tr>
<td>32</td>
<td>Trace traps received</td>
</tr>
</tbody>
</table>

To select multiple trace types, add the individual values. To request a netmon trace value of 2, enter \usr\ov\bin\netmon -M 2. To request netmon values of 2 and 16, enter \usr\ov\bin\netmon -M 18.

**Diagnosing Problems with the NetView Program**

The components of the NetView program use a lot of memory, swap, page, and disk space, so some problems can be caused by resource exhaustion on the management system. See [Chapter 1. Installing the NetView Program](#) for the recommended configuration. The performance of the NetView program is affected primarily by these factors:

- The amount of memory installed in the management system
- The management system’s processor (CPU) speed
- The performance of the Microsoft Windows operating system
- The amount of network traffic
- The number of processes running
- The size of the topology database

If you suspect a problem with the NetView program, and if an equivalent Windows system command exists for the intended operation, exit the NetView program completely and try the command in a Windows command prompt window. If the operation is successful at the command line, the NetView program is probably at fault. If so, the following sections can help you diagnose specific NetView problems and keep them from recurring.
Checking Processes

If the NetView program does not start when you double-click on the NetView icon in the program group, follow these steps:
1. Double-click the Daemon Status icon to check the status of the NetView daemons. They may not be running.
2. If any of the daemons are not running, stop all the daemons by double-clicking Stop Daemons.
3. Double-click Process Viewer icon to see if any daemons are still running. If any are, double-click Process Viewer to stop them.
4. Double-click the Start Daemons icon to restart all the daemons.

Submap Printing Problems with non-True Type Fonts

Submap printing does not display label text if a non-True Type font is selected as the object label font. To change to a TrueType font:
- Select Options→Console Settings.
- Click Object Label Font....
- Select a font from the list that has the True Type markings to the left.
- Click OK.
- Click OK.

Checking NetView Paging Space

When too much paging space is being used, NetView performance might degrade. Use the Performance Monitor application under the Windows Administrative Tools program group to determine which user or application is using the most paging space. If you do not have enough, increase the paging space:
1. Double-click the Control Panel icon in the Windows Main program group.
2. Double-click the System icon.
3. Click Virtual Memory to adjust the paging space.

ovwdb Storage Capacity

The ovwdb daemon increases memory storage space dynamically and is only limited by the amount of memory available. The option to set the number of objects ovwdb can handle is now obsolete.

Managing Hardware and Software

To maintain the network and to optimize its performance, the hardware and software requirements of the NetView program must be met. See "Chapter 1. Installing the NetView Program" on page 1 for a list of software and hardware requirements and for information about configuration options.

Hardware and Software Checklist

To avoid performance problems with the NetView program, follow this checklist:
- Ensure that \usr\ov\bin is in the system environment variable PATH.
- Install additional memory to increase NetView performance.
- Make sure the management system meets the hardware and software prerequisites. Consider using the configuration options available for the NetView program and the Windows SNMP Agent.
Backing Up Files
When you back up original files, do not copy the files to a new file name in the
same directory. All files in the following directories are considered "live;" that is,
every file in these directories is used and you will create duplicate definitions when
you create duplicate files.

\usr\ov\registration\*
\usr\ov\fields\*
\usr\ov\symbols\*
\usr\ov\bitmaps\*
\usr\ov\conf\*
\usr\ov\help\*

Therefore, to back up your original files, do one of the following:
• Create a directory called \usr\ov\backup and replicate the directories listed above
  in this directory.
• Back up the original files. Back up all database files together.

Maintaining the Data Collection Directory
The snmpCollect daemon might use up too much space in the SQL database. To
prevent this problem, follow these steps:
1. Reduce the polling intervals.
   Select Tools..MIB..Collect Data from the NetView main menu to decrease the
   collection interval. If you want to check only thresholds, do not store the data.
2. Control the amount of MIB data you are collecting and set reasonable collection
   intervals for the netmon daemon. Collecting too much MIB data or setting
   intervals that poll too frequently can use up too much of the NetView program
   host processor and network resources.
   To control the polling intervals, select Options..Polling from the NetView main
   menu.
3. Set an at entry to periodically remove SNMP data entries in the SQL database.
   Select Options..Server Setup, the Files tab page, then Schedule
   snmpCollect Data to Delete.

Solving Remote Operation Problems
If you are having problems running an operation to a remote node, follow this
checklist:
• If you configured agent community names, verify that the node’s community
  name is in the \usr\ov\conf\ovsnmp.conf file and matches the community name
  configured on that node. Select Options..SNMP from the NetView menu to
  change community names.
• If you are having problems with the Event Browser, check:
  – Performance of the Current Events application suffers if the performance of
    the entire network is poor. For example, if the name server for your network
    takes a long time to respond, the response time of the NetView program is
degraded. Ensure that the NetView configuration and the name server
    configuration are properly adjusted.
    If you are using a domain name server (DNS), and the system goes down,
you will not have access to the network. You can set up the management
system as a domain name server:
  1. Double-click the Windows Control Panel icon.
  2. Double-click the Network icon.
  3. Select TCP/IP Protocol from the list of installed software.
4. Click **Configure**.
5. Click **Connectivity**.
6. Add the domain name server.

Or, you can specify the device and internet address in your `\<winnt>\system32\etc\drivers\hosts` file. Refer to [Chapter 9. Resolving Network Problems](#) on page 99 for help with diagnosing network performance problems. Stop the Event Browser by selecting **File..Exit**. Then restart the Event Browser by selecting the **Events** icon from the NetView Application toolbar or by selecting **Monitor..Events..All** from the main menu.

- Clear the `\usr\ov\log\trapd.log` file by selecting the **Server Setup** options. All previous events are lost when you clear this file.
- Check the events application by using the Windows Performance Monitor application in the Administrative Tools program group to determine if there are any paging space performance problems.
- Check the NetView **Options..Events..Trap Settings** dialog to ensure that **Only Log As Event** is selected from the **Display the Trap As** combo box. For more information, see [Ftrapd Daemon](#) on page 133.

- **Select Test..Locate Route** from the NetView menu. If this operation fails, verify that the nodes in the route support SNMP.
- Ensure that the remote node supports the operation. Most of the NetView program's operations require SNMP-based, MIB-I (RFC 1156) or MIB-II (RFC 1213) compliant agent software on the remote node.

## Solving Startup Problems

If you have trouble starting the NetView program, check the use of the `\usr` directory. Use the Windows File Manager to ensure there is enough free space.

For other startup problems, check the startup scripts and the local registration files. Also, see [Chapter 1. Installing the NetView Program](#) on page 1 and the release notes accessed by double-clicking on the Read Me icon in the NetView program group.

## Solving Runtime Problems

If you are having problems running the NetView program, verify the following:

- The release versions of the software programs on your system meet the requirements of the NetView program. For more information, see [Chapter 1. Installing the NetView Program](#) on page 1.
- The file permissions of the NetView data files allow access to users and applications. The following apply:
  - All the files in the `\usr\ov\databases\snmpCollect` directory need to have write permission.
  - You can restrict the permissions on NetView data files by using the MS DOS `attrib` command. For example, you could enter `attrib +r filename` in a command prompt window to make a file read-only.

## Solving Background Graphics Problems

You might have problems with background graphics on your submap views if the images use a large number of colors. Because an 8-bit plane display device supports only 256 colors, loading several images (especially scanned photos) that use a large range of colors might cause the colors in the graphics to be inaccurate.
The NetView program loads all bitmap (.BMP) files, but for best results, use drawn images or graphics with fewer colors.

Submap Symbol Display Problems
The Matrox Millenium Display Driver 2.2. may cause problems to the display of NetView symbols. Use revision level 3.1 or higher to correct the problem.

One or More nvsniffer applications Never Complete
If your system has more nvsniffer applications running that you expect, it is possible that you have scheduled the nvsniffer application to run too frequently and each nvsniffer is taking longer than expected to complete. It has been observed that some nodes, particularly connector devices, can take several minutes to respond to raw port test that the nvsniffer application performs. The problem is magnified if the nvsniffer configuration file being used contains a large number of entries.

If you are using one large nvsniffer configuration file, consider creating several small configuration files and schedule a separate nvsniffer application for each configuration file.

You can also change the frequency of each nvsniffer invocation, and/or specify a larger value for the -r switch when automatic reinvocation of the nvsniffer application is desired. To change the frequency of automatically rescheduled nvsniffer invocations, enter the at command at a command prompt; for each nvsniffer entry that you wish to change, enter:

\[ \text{at <job_ID> /delete} \]

where <job_ID> is the ID of the scheduled nvsniffer command. This will remove the scheduled nvsniffer application from the schedule service. To reschedule the nvsniffer application to run at less frequent intervals, enter the desired nvsniffer command at a command prompt using a larger value for the -r switch. The nvsniffer applications will begin executing at that instant and will automatically reschedule itself at the newly specified frequency.

To force all currently running nvsniffer applications to stop, exit the NetView console and enter the following commands at a command prompt:

\[ \text{ovstop nvcold} \]
\[ \text{ovstart nvcold} \]
\[ \text{netview} \]

The nvsniffer application depends on the nvcold daemon. When the nvcold daemon is stopped, each running nvsniffer application automatically terminates itself.
Chapter 11. Maintenance

As a NetView administrator, plan to perform maintenance, tasks occasionally. The tasks you will need to perform will depend on the needs of your site. This section describes the tasks.

Changing the IP Address or Name

The following two sections describe the actions you must perform if the IP address or host name of the NetView server or client changes.

Changing the Server IP Address or Name

If you have installed NetView in the Single User mode or Server you only need to check the file \%SYSTEMDIR\%system32\drivers\etc\hosts to ensure that the entries are correct for the management station.

If you have NetView clients connected to this NetView server, check the following on each client:

- Ensure that the client can resolve the new IP address and name. You can check this using the host.exe utility. Add an entry to the \%SYSTEMDIR\%system32\drivers\etc\hosts file for the new address and name of the server.
- Use the Registry Editor to modify the ServerName and ServerAddress entries in HKEY_LOCAL_MACHINE\SOFTWARE\Tivoli\NetView\CurrentVersion.
- Ensure that the NetView share is still mounted correctly and is assigned the drive letter indicated by the system variable NV_MOUNT_DRIVE.

Changing the Client IP Address or Name

If the NetView client changes its IP address or name, ensure that the NetView server can resolve the new name and address and derive the correct NetBIOS name. Use the host.exe utility to check this and make a new entry (or modify the existing one) in the file \%SYSTEMDIR\%system32\drivers\etc\hosts on the NetView server machine.

Maintaining NetView Log Files

To diagnose problems with the NetView program that occur while the program is running, you need to know the status of the NetView daemons. If you find out that a daemon is not running, you can restart it and any other daemons on which it depends. The following topics describe how to check daemon status and restart a daemon that is not running properly.

NetView Log File

The file \usr\ov\log\nv.log is the NetView product–specific log file for NetView servers and clients. Information and as error conditions are logged in this file. If you report a problem, the Tivoli customer support team will want to view this file.

Because this file grows over time, rename it periodically, perhaps keeping only one renamed version.

If the process management daemon, ovspmd, was started with the -V option, this status report is much more extensive, including everything the ovspmd daemon knows about the daemons.
Events Log File
Events are stored in the SQL database by default. To prevent the SQL database from growing too large, the trapd daemon periodically removes the oldest events from the database. You can control how many events are preserved and scheduled when, or if, purging should occur. To do this, select Options→Server Setup and use the trapd daemon page. You can also, optionally, specify that events should be written to a log file, \usr\ov\log\oldevents.log, before being purged from the database. Because this file grows over time, you need to have a policy on how to archive this file. This option is only available on a NetView server.

Client/Server Map Events Log File
The map events corresponding to manage, unmanage, acknowledge, or unacknowledge are used to update the icon status colors on each map in a NetView client/server environment and are stored in the file \usr\ov\databases\openview\playitagain.sam on the NetView server.

When the NetView console starts, it reads this file and processes any events it may have missed since the last time it was running.

If you are running NetView clients, this file will grow over time on the NetView server. There is no reason to archive this file. However, before you delete it, ensure that all the client maps are up to date.

MS Access Database for Events
The NetView installation uses a Microsoft Access database to store events if the SQL Server is not present. This Microsoft Access database can become fragmented causing a decrease in performance and an increase in the use of disk space. The trapd daemon now automatically compacts the Events and SnmpCollect databases on startup. You can manually compact the databases by stopping the NetView Service and from tess_ds and snmpcollect_ds select: Start→Settings→Control Panel→ODBC→System DSN→Configure→Compact.

SNMP Data Collection
By default, SNMP data that is collected is stored in the SQL database, which grows over time. To prevent this database from growing too large, a process runs each night that removes the oldest data from the database. To control how many records are preserved and schedule when, or if, purging should occur, select Options→Server Setup and then select the Schedule SnmpCollect Data to Delete pull-down on the Files page.

The installation creates an at scheduler entry for the snmpodump -x5000 command. NetView will normally maintain the scheduler queue by deleting or replacing entries. You can check this by entering the at command at the command prompt to view scheduler entries.

Changing Locale Settings on NetView Servers or Clients
If you choose to change from a non-U.S. English locale on a NetView server to a different non-U.S. English locale, also ensure that each NetView client is deinstalled and reinstalled for the same locale. There is no migration for locale changes. This action requires that you are running the appropriate language-specific edition of Windows on each machine.
Appendix A. System Performance Considerations for Large Networks

If your network contains more than 4000–5000 nodes, check the following preventative maintenance items, periodically, to keep your system running efficiently:

- Upon startup, the NetView graphical user interface can take several minutes to update with the new default settings for status propagation. The computation of the status of a node includes the status of Service objects that the nvsniffer application has discovered. The higher the number of nodes in the network, the longer it can take to compute propagated status. This is evident if the update map status is displayed in the NetView status bar for several minutes after starting the NetView graphical user interface. If you do not require the status of service objects to contribute the overall status of a node, you can change the default settings for this property using Edit→Properties→Map→Properties→IPmap→Properties and uncheck Propogate Status of Services to Nodes.

- Over time, disk fragmentation can degrade system performance, especially with larger networks. It is recommended that you periodically defragment the disk on which the NetView product is installed, and the system disks. Larger networks will require more frequent fragmentation of the NetView partition. This process can be easily automated using the Schedule service (AT command) to set up automatic jobs to stop the NetView daemons, execute the defragger, and then start the NetView daemons and GUI. Defragmentation software generally works best when there are no applications running.

- If the disk on which the NetView product is installed is running low on available space, there are a few files that may have grown excessively. In particular, events and SNMP data collections (if they are stored in Access databases) can grow to hundreds of MB. It is recommended that you periodically compress these Access databases. The trapd daemon automatically compacts the Events and SnmpCollect databases on startup. You can manually compact the databases by stopping the NetView Service and for tess_ds and snmpcollect_ds select. Start→Settings→Control Panel→ODBC→System DSN→Configure→Compact

Other files, such as log files, should also be check periodically (see Chapter 11 Maintenance on page 139).

- As the size of the network grows, the default Polling and SNMP options may cause the netmon daemon to become back-logged with ICMP and SNMP requests. The Network Monitor Activity bars located at the bottom right of the NetView graphical user interface and the Network Monitor Workload indicator in the Options→Polling dialog indicate the current workload for netmon. The Polling Options dialog also indicates when the next scheduled Configuration poll and New Node polls should occur. If either of these values have fallen behind or are overlapping with each other, consider scheduling these activities to occur at different times of the day. For example, the default value for Poll to Discover New Node is set to occur at automatic intervals; changing this value to an absolute daily time (late night or early morning) can prevent excessive simultaneous system and network activity.

To determine how far behind in sending ICMP messages the netmon daemon is at a given instant, monitor the NetView perform values for NETMON:PingsBehind and NETMON:SNMPsBehind. To start the function, enter the perform command at a command prompt, then select the NetView: Edit→Add to Chart....
Select **NetView** in the Object pulldown menu, then select **NETMON:PingsBehind** and **NETMON:SNMPsBehind** from the Counter selection list and click **Add**. You may want to adjust the sampling frequency using the **Options>Chart** menu option.

If you determine that netmon is falling behind with pings (ICMP messages) and the Discovery Ping Spray option is enable, select **Options>Server Setup>Discovery** to disable ping spray until netmon catches up; the cached information may be temporarily exceeding its storage limits. Ping Spray can be resumed when netmon has caught up.

The values for Ping Timeouts can also affect the ability of the netmon daemon to keep up with the requested workload, so be sure that these values are reasonable with respect to the network load and performance.

- More system memory will help sustain adequate system performance as the number of discovered nodes increases; be sure to adjust the size of Virtual Memory as you increase the amount of physical memory (additional disk space may be required to accommodate the additional paging file requirements). The following are some recommendations:
  - Up to 4,000 nodes: 128 MB physical memory, 256 MB paging file minimum
  - 4,000–8,000 nodes: 256 MB physical memory, 512 MB paging file minimum
  - 8,000 or more nodes: Consult your Tivoli representative for a recommendation.

- Avoid creating SmartSets that will contain 2000 or more members. If you accidentally create such a SmartSet you may notice the following symptoms:
  - Displaying the large SmartSet in the NetView graphical user interface may cause the console to use 100% of the CPU.
  - The NetView graphical user interface (console) may be suspended.

To solve this problem, stop the NetView graphical user interface and enter **smartsetutil D XXX** at a command prompt, where **XXX** is the name of the large SmartSet.

- It has been observed that with very large databases (8000 or more nodes) the ovwdb daemon can stop functioning normally if the database files it uses to store objects becomes too fragmented. This becomes evident when the other daemons or applications appear to freeze up or fail to connect to the ovwdb daemon. Most notably, the ovtopmd daemon and the snmpcollect daemons have difficulty starting due to the amount of time it takes to synchronize with the ovwdb daemon. If this condition occurs on your system, examine the size of the following file:
  \usr\ov\databases\openview\owdb\current\value_info.pag

If this file is more than 50 MB, use the Windows Explorer compress feature to compress all files under \usr\ov\databases\openview. To do this, locate this directory in the Windows Explorer view, click on the directory, select **Properties** and on the General tab page check the **Compress** checkbox, then click **OK** or **Apply**.

This procedure may need to be performed periodically to avoid this problem.

For daemons that fail to start properly or for those daemons that take a long time to start, you can start the core NetView daemons individually by entering the following sequence of commands at a command prompt:
ovstop
ovstart ovwdb
ovstart trapd
ovstart ovtopmd
ovstart
Appendix B. Installing NetView Using Microsoft System Management Server (SMS)

This section describes how to install NetView using the Microsoft System Management Server (SMS) and new NetView Installation Command Line Options to Facilitate Installation.

Steps to Install NetView using SMS

The NetView kit includes a Package Definition File (.pdf file) for NetView. The following steps describe how to install NetView using SMS:

1. Place the NetView distribution media in a drive that can be shared with the SMS client systems. Alternatively, you may copy the entire contents of the NetView distribution media to a drive in your network that can be shared with the SMS client systems. These instructions will refer to the root directory on the NetView distribution media or the root directory of the copy of the NetView distribution media as the NetView kit directory.

2. From the SMS Administrator on the SMS server system, create a SMS package for NetView. Import the netview.pdf file that is located in the NetView kit directory.

3. Next, click Workstations..., select the type of installation and select the path that the clients will use to copy the kit files. The path should be available to the SMS client systems and should be in UNC format.

4. If you want to specify any additional command line options to the installation program, click Properties... in the Setup Package for Workstations dialog box and edit the installation command line. For more information about the NetView installation command line switches, refer to "Using NetView Installation Command Line Options to Facilitate Installation via SMS (Alpha Platform)" on page 146.

5. Create a distribution job for the NetView installation package by dragging the package to machines in the domain.

If you are installing a NetView client, the properties of the client package setup, as described in Step 4 above, specify the IP address of the NetView server that the installed client will use. Additionally, if you specified a password for the NetView service account when the NetView server installation was performed, you must specify the same password as a parameter to the NetView client installation so that the client can access the server. Refer to "Using NetView Installation Command Line Options to Facilitate Installation via SMS (Alpha Platform)" on page 146 for more information on installation command line switches.

Creating a Silent Installation to Facilitate Installation via SMS (Intel Platform)

The NetView installation program provides a mechanism for recording the settings used during an installation. The recorded settings can then be used to perform unattended or silent installation.

To utilize this capability, execute the NetView installation with the -r option. All of the settings that are entered during the installation are recorded in a response file called setup.iss. To use the settings for another installation, execute the NetView installation with the -s option. You may also use the -f1 option to specify the name
and location of the response file and the -f2 option to specify the location of the silent installation log file that will be generated (this is not the same as the NetView installation log).

It is important to note that some of the dialog boxes during an installation depend on system settings, or on selections made earlier in the installation. For example, a fresh install will ask which drive to install on, and what type of setup to perform. An upgrade install will not ask either of these questions, since it knows what already exists. In this case, you will not be able to use the response file from an Upgrade install on a computer that has never had NetView installed on it.

To verify if a silent installation succeeded, look at the ResultCode in the [ResponseResult] section of the silent installation log file (setup.log).

### Table 25. Silent Install ResultCode and Error Table

<table>
<thead>
<tr>
<th>ResultCode</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Success</td>
</tr>
<tr>
<td>1</td>
<td>General Error</td>
</tr>
<tr>
<td>2</td>
<td>Invalid mode</td>
</tr>
<tr>
<td>3</td>
<td>Required data not found in Setup.iss file</td>
</tr>
<tr>
<td>4</td>
<td>Not enough memory available</td>
</tr>
<tr>
<td>5</td>
<td>File does not exist</td>
</tr>
<tr>
<td>6</td>
<td>Cannot write to the response file</td>
</tr>
<tr>
<td>7</td>
<td>Unable to write to the log file</td>
</tr>
<tr>
<td>8</td>
<td>Invalid path to the response file</td>
</tr>
<tr>
<td>9</td>
<td>Not a valid list type (string or number)</td>
</tr>
<tr>
<td>10</td>
<td>Data type is invalid</td>
</tr>
<tr>
<td>11</td>
<td>Unknown error during setup</td>
</tr>
<tr>
<td>12</td>
<td>Dialogs are out of order</td>
</tr>
<tr>
<td>51</td>
<td>Cannot create the specified folder</td>
</tr>
<tr>
<td>52</td>
<td>Cannot access the specified file or folder</td>
</tr>
<tr>
<td>53</td>
<td>Invalid option selected</td>
</tr>
</tbody>
</table>

### Using NetView Installation Command Line Options to Facilitate Installation via SMS (Alpha Platform)

The NetView installation program provides several command line options that facilitate installation when using SMS. The following table lists the names of the relevant options and describes each option. It is important to note the command line options that accept a value. The value must be specified immediately following the option name without intervening white space.

### Table 26. NetView Installation Command Line Options and Definitions

<table>
<thead>
<tr>
<th>Command Line Options</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>–nq</td>
<td>Do not prompt for installation options. Use this option to perform an unattended installation.</td>
</tr>
<tr>
<td>Command Line Options</td>
<td>Definition</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------</td>
</tr>
<tr>
<td>–up password</td>
<td>Specify the password for the NetView service account that is created by a server or standalone installation or use this option to specify the password for the NetView server file share in a client installation.</td>
</tr>
<tr>
<td>–un username</td>
<td>The username under which this software will be registered. If this option is not supplied, the default username established on the SMS client will be used.</td>
</tr>
<tr>
<td>–uo organization</td>
<td>The organization name under which this software will be registered. If this option is not supplied, the default company name established on the SMS client will be used.</td>
</tr>
</tbody>
</table>
| –dm discover_mode    | Use this option to specify discover mode:  
0–discover only local networks  
1–discover all networks |
| –dd install_directory| The install directory on the SMS client system can be supplied with this option. If the –dd command line option is omitted, the NetView installation procedure examines the SMS client local hard disks and installs NetView on the first local hard disk that has sufficient space. |
| –im install_mode     | Use this option to specify the installation mode:  
0–standalone NetView installation  
1–client NetView installation  
2–server NetView installation |
| –ha host_address     | If performing a NetView client installation, use this command line switch to specify the host address of the NetView server system. If the –hn option is supplied, it is not necessary to specify the –ha switch. |
| –hn host_name        | If performing a NetView client installation, use this command line switch to specify the name of the NetView server system. |
Glossary

This glossary defines technical terms used in the documentation for Tivoli products and includes selected terms and definitions from:

- The American National Standard Dictionary for Information Systems, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The ANSI/EIA Standard—440-A, Fiber Optic Terminology. Copies may be purchased from the Electronic Industries Association, 2001 Pennsylvania Avenue, N.W., Washington, DC 20006. Definitions are identified by the symbol (E) after the definition.
- The Information Technology Vocabulary developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.
- Internet Request for Comments: 1208, Glossary of Networking Terms
- Internet Request for Comments: 1392, Internet Users’ Glossary

The following cross-references are used in this glossary:

Contrast with:

This refers the reader to a term that has an opposed or substantively different meaning.

See:

This refers the reader to (a) a related term, (b) a term that is the expanded form of an abbreviation or acronym, or (c) a synonym or more preferred term.

Obsolete term for:

This indicates that the term should not be used and refers the reader to the preferred term.

A


absolute path. A path that begins with the root directory. The absolute path may also be known as the “full pathname.” Contrast with relative path.

abstract model. In Tivoli Global Enterprise Manager, the business description files that logically describe a particular business system.

abstract syntax notation 1 (ASN.1). The Open Systems Interconnection (OSI) method for abstract syntax specified in the following standards:

accelerator. (1) In a user interface, a key or combination of keys that invokes an application-defined function. (2) In the AIX® Toolkit, a keyboard alternative to a mouse button action; for example, holding the <Shift> and <M> keys on the keyboard can be made to post a menu in the same way that a mouse button action does. Accelerators typically provide increased input speed and greater convenience.

access control. In computer security, the process of ensuring that the resources of a computer system can be accessed only by authorized users in authorized ways.

access control list. (1) In computer security, a collection of all access rights for one object. (2) In computer security, a list associated with an object that identifies all the subjects that can access the object and their access rights; for example, a list associated with a file that identifies users who can access the file and identifies their access rights to that file.

ACF. See Adapter Configuration Facility.

action. (1) An operation on a managed object, the semantics of which are defined as part of the managed object class definition. (2) In the AIX® operating system,
a defined task that an application performs. An action modifies the properties of an object or manipulates the object in some way.

**Action Message Retention Facility (AMRF).** An OS/390 facility that, when active, retains all action messages except those specified by the installation.

**adapter.** (1) A part that electrically or physically connects a device to a computer or to another device. (2) Software that enables different software components or products to interact with one another. (3) See **event adapter**.

**Adapter Configuration Facility (ACF).** In the Tivoli Enterprise Console, a graphical user interface that enables a Tivoli administrator to easily configure and customize event adapters.

**ADE.** See [Tivoli Application Development Environment](#).

**ADF.** See [application description file](#).

**Administrative Domain.** A collection of hosts and routers, and the interconnecting networks, managed by a single administrative authority.

**administrator.** See [Tivoli administrator](#).

**administrator collection.** In a Tivoli environment, the collection for administrator objects that is generated by Tivoli Enterprise™ software. This container is represented by the Administrator icon on the Tivoli desktop; opening the icon provides access to information about each Tivoli administrator.

**admin role.** See [authorization role](#).

**Advanced Business Application Programming/4 (ABAP/4).** A fourth-generation programming language in which SAP R/3 application software is written.

**AEF.** See [Tivoli Application Extension Facility](#).

**agent.** (1) In systems management, a user that, for a particular interaction, has assumed an agent role. (2) An entity that represents one or more managed objects by (a) emitting notifications regarding the objects and (b) handling requests from managers for management operations to modify or query the objects. (3) A system that assumes an agent role.

**Agent Policy Manager (APM).** In Tivoli NetView, a function that controls Mid-Level Manager (MLM) configurations in a network from a single, central location.

**agent role.** In systems management, a role assumed by a user in which the user is capable of performing management operations on managed objects and of emitting notifications on behalf of managed objects.

**aggregate object.** In the NetView Graphic Monitor Facility, an object that represents a collection of real objects.

**AIXwindows Toolkit.** An object-oriented collection of C language data structures and subroutines that supplement the Enhanced X-Windows Toolkit and simplify the creation of interactive client application interfaces.

**alarm.** A signal, either audible or visual, at a device such as a display station or printer that is used to notify the user that a condition requiring the user's attention exists.

**alarm level.** In Tivoli Distributed Monitoring, the state of a monitor when a specified threshold has been reached. A Tivoli administrator can set thresholds for each alarm level and have Tivoli Distributed Monitoring trigger a different response (an action and an event) for each level. There can also be several responses for each alarm level.

**alert.** (1) A message sent to a management services focal point in a network to identify a problem or an impending problem. (2) In SNA management services (SNA/MS), a high priority event that warrants immediate attention.

**alias name.** A name that is defined in one network to represent a logical unit name in another interconnected network. The alias name does not have to be the same as the real name; if these names are not the same, translation is required.

**alias name translation facility.** In Tivoli NetView for OS/390, a function for converting logical unit names, logon mode table names, and class-of-service names used in one network into equivalent names to be used in another network.

**allomorphism.** The ability of an instance of a class to be managed as an instance of one or more different but compatible managed object classes.

**AMP.** See [application management package](#).

**AMRF.** See [Action Message Retention Facility](#).

**AMS.** See [Application Management Specification](#).

**AOF.** See [application object file](#).

**AON.** See [Automated Operations Network](#).

**APAR.** See [authorized program analysis report](#).

**API.** See [application programming interface](#).

**APM.** See [Agent Policy Manager](#).

**application.** A collection of software components used to perform specific types of user-oriented work on a computer.
application description file (ADF). In the context of the Application Management Specification (AMS), a readable, ASCII text file that contains information for managing an application. Application description files are based on the Management Information Format (MIF). Application description files include component description files, global description files, and business description files (business system description files, business system component description files, business system mapping description files, and business subsystem description files).

application management package (AMP). In a Tivoli environment, a compressed file that contains the application description files and other necessary files for managing an application. These include one global description file, one or more component description files, task scripts, and executable programs. The application management package can also include the application object file or the source files for the application itself.

Application Management Specification (AMS). A specification that presents a standard for managing applications. The Application Management Specification was developed in collaboration with the Tivoli Partners and Tivoli customers to address the problems associated with multitiered applications.

application object file (AOF). In a Tivoli environment, an ASCII text file that contains the names of the global description file and the component description files, which together describe the management characteristics of an application. The Tivoli Module Designer and the Tivoli Module Builder can import an application object file that was created by the obsolete Tivoli Developer Kit.

application plane. In Tivoli NetView, the submap layer on which symbols of objects that are managed by at least one network or systems management application program are displayed. Symbols on the application plane are displayed without shading, which makes them appear directly against the background plane. See used plane.

application programming interface (API). A software interface that enables applications to communicate with each other. An API is the set of programming language constructs or statements that can be coded in an application program to obtain the specific functions and services provided by an underlying operating system or service program.

application registration file (ARF). A file created to integrate an application program into Tivoli NetView by defining (a) the application program's position in the menu structure for Tivoli NetView, (b) where help information is found, (c) the number and types of parameters allowed, (d) the command used to start the application program, and (e) other characteristics of the application program.

Application Response Measurement (ARM). An application programming interface that was developed by a group of leading technology vendors, including Tivoli Systems Inc., and that can be used to monitor the availability and performance of business transactions within and across diverse applications and systems. The monitoring is done from the perspective of the applications; therefore, it reflects the units of work that are important from the perspective of the business. For example, using ARM, a business could instrument an application to discover:

- Whether the application is hung
- The level of response time that the application is experiencing
- Where the bottlenecks are occurring during the execution of the application
- Who is using the application and how much they are using it
- How to tune the system environment to run the application more efficiently
- What the application is doing during the reported response time
- Where in the system environment a transaction is spending its time

APPNTAM. See SNA topology manager.

APPN® Topology and Accounting Manager (APPNTAM). See SNA topology manager.

ARF. See application registration file.

ARM. (1) See Application Response Measurement. (2) See automatic restart manager.

ARM agent. An agent that monitors software that is instrumented using the Application Response Measurement (ARM). The ARM agent is shipped as part of Tivoli Distributed Monitoring.

ASN.1. See Abstract syntax notation 1.

ASYN. See asynchronous.

asynchronous (ASYNC). (1) Pertaining to two or more processes that do not depend upon the occurrence of specific events such as common timing signals. (2) Without regular time relationship; unexpected or unpredictable with respect to the execution of program instructions.

asynchronous monitor. In Tivoli Distributed Monitoring, a monitor that receives data in an unsolicited event and interprets the data immediately. Contrast with synchronous monitor.

attribute. A characteristic that identifies and describes a managed object. The characteristic can be determined, and possibly changed, through operations on the managed object.
authentication. (1) In computer security, verification of
the identity of a user or the user’s eligibility to access an
object. (2) In computer security, verification that a
message has not been altered or corrupted. (3) In
computer security, a process used to verify the user of
an information system or protected resources.

authorization. (1) In computer security, the right
granted to a user to communicate with or make use of a
computer system. (T) (2) An access right. (3) The
process of granting a user either complete or restricted
access to an object, resource, or function.

authorization role. In a Tivoli environment, a role
assigned to Tivoli administrators to enable them to
perform their assigned systems management tasks. A
role may be granted over the entire Tivoli Management
Region or over a specific set of resources, such as
those contained in a policy region. Examples of
authorization roles include: super, senior, admin, and
user.

authorized operator. In Tivoli NetView for OS/390, an
authorized operator who has been authorized to receive
undeliverable messages and lost terminal messages. See
authorized receiver.

authorized program analysis report (APAR). A
report of a problem caused by a suspected defect in a
current unaltered release of a program.

authorized receiver. In Tivoli NetView for OS/390, an
authorized operator who receives the unsolicited and
authorized-receiver messages that are not assigned to a
specific operator.

Automated Operations Network (AON). In Tivoli
NetView for OS/390, the component that handles
automated resource monitoring, recovery, and tracking.

automated response. In a Tivoli environment, a
predefined response to particular events that is
automatically executed by a Tivoli application. For
example, if the Tivoli Enterprise Console detects that a
process has terminated early, it can automatically restart
the process without the intervention of the Tivoli
administrator.

automatic reactivation. In Tivoli NetView for OS/390,
the activation of a node from the inactive state without
any action by the network operator.

automatic restart manager (ARM). An OS/390
recovery function that can automatically restart batch
jobs and started tasks after they or the system on which
they are running terminate unexpectedly.

AutoPack Control Center. A Tivoli Software
Distribution tool that is installed on a Windows-based
PC and enables a Tivoli administrator to create an
AutoPack file. The AutoPack Control Center produces
the AutoPack file by (a) taking snapshots of the PC’s
drive and system configuration before and after the
installation of an application on the PC and (b) capturing
the differences between these snapshots and the
distribution instructions in the AutoPack file.

AutoPack file. In Tivoli Software Distribution, an
installable image that is used to distribute
“shrinkwrapped” applications to multiple PC targets. The
file contains a description of PC software application
directories, information on how to distribute
files and directories, and any system configuration
changes needed by the application. A Tivoli
administrator must associate an AutoPack file with an
AutoPack profile.

AutoPack profile. A Tivoli Software Distribution profile
that references an AutoPack file.

autotask. (1) In Tivoli NetView for OS/390, an
unattended operator station task that does not require a
terminal or a logged-on user. Autotasks can run
independently of VTAM® and are typically used for
automated console operations. (2) Contrast with
logged-on operator.

availability management. The Tivoli management
discipline that addresses the gathering, collecting, and
routing of information regarding the operational status of
an organization’s network computing system and
enables the appropriate corrective action. See
deployment management, operations and
administration, and security management.

B

backend. In the AIX operating system, the program
that sends output to a particular device.

background plane. In Tivoli NetView, the lowest
submap layer. The background plane provides the
background against which symbols are displayed. A
background picture can be placed in the background
plane to provide a context for viewing symbols. See
application plane and user plane.

background process. (1) A process that does not
require operator intervention but can be run by the
computer while the workstation is used to do other
work. (2) In the AIX operating system, a mode of
program execution in which the shell does not wait for
program completion before prompting the user for
another command. (3) Contrast with foreground
process.

background task. A task that is running even though
the user is not currently interacting with it. Contrast with
foreground task.

bandwidth. A measure of the capacity of a
communication transport medium (such as a TV cable)
to convey data.
BARC program. Obsolete term for configuration program. "BARC" is an acronym for "before, after, removal, and commit."

BAROC. See Basic Recorder of Objects in C

base module. In a Tivoli environment, a management module that describes the basic management characteristics of a particular application or business system to the Tivoli management software. Unlike Tivoli GEM modules and Tivoli Plus modules, base modules are developed without the use of a template.

bash. Bourne-again shell. A portable, command-line interface and script interpreter that is compatible with the UNIX® Bourne and Korn shells and includes some features of the UNIX C shell.

Basic Input/Output System (BIOS). Code that controls basic hardware operations, such as interactions with diskette drives, hard disk drives, and the keyboard.

Basic Object Adapter (BOA). Software that provides CORBA-compliant services for object implementations.

Basic Recorder of Objects in C (BAROC). In the event server of the Tivoli Enterprise Console, the internal representation of the defined event classes.

basic sequential access method (BSAM). In the NetView Performance Monitor (NPM), the method by which all PIUs collected for selected LUs can be logged into a sequential data set as they pass through VTAM.

BSAM. See basic sequential access method.

business component. An application or other system resource that can be managed by systems management software.

business description file (BDF). In a Tivoli environment, a generic name for any of these application description files: business system description file (BSDF), business system component description file (BCDF), and business system mapping description file (BMDF), and business subsystem description file (BSSDF).

business subsystem description file (BSSDF). In the context of the Application Management Specification (AMS), an optional application description file that enables the logical grouping of business components in a business system. In this file, a Tivoli administrator can specify tasks and monitors that are common to the subsystem. The business subsystem description file references the applicable business system description file and one or more business system component description files.

business system. A group of diverse but interdependent applications and other system resources that interact to accomplish specific business functions.

business system component description file (BCDF). In the context of the Application Management Specification (AMS), an application description file that...
defines the logical representation of a business component. When defining a business system, at least one business system component description file is required for each business system definition.

**business system description file (BSDF).** In the context of the Application Management Specification (AMS), the highest-level application description file that identifies the components of a business system, including monitors, tasks, and connections. A Tivoli administrator can also define icon and help files at this level (for creating a business system icon and help information).

**business system mapping description file (BMDF).** In the context of the Application Management Specification (AMS), the application description file that maps a real business component (which is defined in a component description file) to a logical business component (which is represented in a business system component description file). Each business system mapping description file references a business system component description file.

**CC.**

**CADAM.** Computer-Aided Design and Manufacturing. The use of computers in the design and manufacture of products such as cars, airplanes, ships, and computers.

**call.** (1) The action of bringing a computer program, a routine, or a subroutine into effect, usually by specifying the entry conditions and jumping to an entry point. (2) In data communication, the actions necessary to make a connection between two stations on a switched line. (3) In communications, a conversation between two users. (4) To transfer control to a procedure, program, routine, or subroutine. (5) To attempt to contact a user, regardless of whether the attempt is successful.

**callback.** In the AIX operating system, a procedure that is called if and when certain specified conditions are met.

**canonical.** In computer science, pertaining to an expression that conforms to a specific set of rules.

**CATIA.** Computer-Graphics Aided Three-Dimensional Interactive Application.

**CC.** See [change control](#).

**CCMS.** (1) See [Computing Center Management System](#). (2) See [Configuration Change Management System](#).

**CDF.** See [component description file](#).

**CDNM session.** See [cross-domain network managed session](#).

**CDS.** See [control data set](#).

central site control facility (CSCF). In Tivoli NetView for OS/390, NetView for VM, and NetView for VSE, a function that allows a network operator to execute the test facilities of the IBM 3172 Nways® Interconnect Controller and the IBM 3174 Establishment Controller remotely from the NetView console.

**change control (CC).** The use of change management commands for the installation or removal of software or data.

**change control administrator.** A person responsible for software distribution and change control activities.

**change control client.** A workstation that (a) receives software and data files from its change control server and (b) installs and removes software and data files as instructed by its change control server.

**change control domain.** A change control server and its change control clients.

**change control server.** A workstation that controls and tracks the distribution of software and data files to other workstations.

**change control single node.** A workstation that controls, tracks, installs, and removes software and data files for itself. A CC single node can also prepare software for distribution. Contrast with change control client and change control server.

**change management.** The process of planning (for example, scheduling) and controlling (for example, distributing, installing, and tracking) software changes over a network. This is sometimes known as "software management."

**check box.** A square box with associated text that represents a choice. When a user selects the choice, the check box is filled to indicate that the choice is selected. The user can clear the check box by selecting the choice again, thereby deselecting the choice.

**checkpoint.** (1) Information about the status of a program's execution or the status of a data transfer that is recorded to enable the program or the data transfer to be restarted if it is ever interrupted. (2) The time at which such information is recorded. (3) To record such information.

**child process.** In the UNIX operating system, a process, started by a parent process, that shares the resources of the parent process. See [fork](#).

**child resource.** In the NetView Graphic Monitor Facility, a resource that is directly subordinate to another resource (the parent) in a hierarchy.

**CICS.** See [Customer Information Control System](#).

**class.** (1) In object-oriented design or programming, a model or template that can be instantiated to create
objects with a common definition and therefore, common properties, operations, and behavior. An object is an instance of a class. (2) In the AIX operating system, pertaining to the I/O characteristics of a device. System devices are classified as block or character devices.

CLI. See command line interface.

client. A computer system or process that requests a service of another computer system or process that is typically referred to as a server. Multiple clients may share access to a common server.

client daemon. An AIX process that performs the client’s operations.

client/server. In communications, the model of interaction in distributed data processing in which a program at one site sends a request to a program at another site and awaits a response. The requesting program is called a client; the answering program is called a server.

client workstation. In the NetView Graphic Monitor Facility, a workstation that depends on a server workstation to provide it with views and status information. A client workstation receives status information from the server workstation over an LU 6.2 session.

cloning. (1) In a Tivoli environment, an operation that enables a Tivoli administrator to replicate profiles. This capability simplifies the task of creating multiple profiles with similar properties. See prototype profile. (2) In a Tivoli environment, a function of Tivoli NetView for OS/390 that enables a system programmer to replicate NetView definitions across the systems comprising a sysplex, thus simplifying the task of creating multiple NetView definitions with similar properties.

CNM. See communication network management.

CNM application program. A VTAM application program that issues and receives formatted management services request units for physical units. Tivoli NetView for OS/390 is an example of a CNM application program.

CNM processor. In Tivoli NetView for OS/390, a program that manages one of the functions of a communication system. A CNM processor is executed under control of Tivoli NetView for OS/390.

collaborative management. A cooperative relationship between Internet commerce partners and Internet service providers (ISPs) to ensure the successful completion of business transactions.

collection. In a Tivoli environment, a container that groups objects on a Tivoli desktop, thus providing the Tivoli administrator with a single view of related resources. Either the Tivoli Management Framework or a Tivoli administrator can create a collection. The contents of a collection are referred to as its members. Examples of collections include the administrator collection and the generic collection; the administrator collection is an example of a collection generated by the Tivoli Management Framework.

collection point block (CPB). In the NetView Performance Monitor (NPM), a control block used to coordinate the collection of network and session data.

combined alert. In Tivoli NetView for OS/390, an alert that includes elements of a non-generic and a generic alert in one network management vector transport (NMVT).

command. (1) A request from a terminal for the performance of an operation or the execution of a particular program. (2) In Tivoli NetView for OS/390, a sequence of characters that is submitted to cause an action. A command contains a verb and an object.


command facility. In Tivoli NetView for OS/390, the component that is a base for command processors that can monitor, control, automate, and improve the operation of a network.

command indicator. In the NetView Graphic Monitor Facility, a numeric identifier that is assigned to a network resource by its controlling resource manager to indicate the command support characteristics for the resource.

command interpreter. In the AIX operating system, a program that sends instructions to the kernel.

command line interface (CLI). A type of computer interface in which the input command is a string of text characters. Contrast with graphical user interface.

command list. In Tivoli NetView for OS/390, a list of commands and statements designed to perform a specific function for the user. Command lists can be written in REXX or in the NetView command list language.

command procedure. In Tivoli NetView for OS/390, a command list, a command processor written in a high-level language (HLL), or a NetView pipeline.

command processor. In Tivoli NetView for OS/390, a module designed to perform a specific function for the user. Users can write command processors in assembler language or in a high-level language (HLL); command processors are invoked as commands.
command profile editor (CPE). In Tivoli Global Enterprise Manager and Tivoli NetView for OS/390, a function of the topology console that enables Tivoli administrators who have the proper administrative authority to control the content, order, and capabilities of pop-up menus for individual operators or groups of operators.

commit operation. In Tivoli Software Distribution, an operation performed by a configuration program on target managed nodes after a file package distribution. This function enables a Tivoli administrator to distribute a file package to multiple targets and to make the distributed information available on all targets at the same time.

Common Object Request Broker Architecture (CORBA). A specification produced by the Object Management Group (OMG) that presents standards for various types of object request brokers (such as client-resident ORBs, server-based ORBs, system-based ORBs, and library-based ORBs). Implementation of CORBA standards enables object request brokers from different software vendors to interoperate.

Common Programming Interface for Communications (CPI-C). An evolving application programming interface (API), embracing functions to meet the growing demands from different application environments and to achieve openness as an industry standard for communications programming. CPI-C provides access to interprogram services such as (a) sending and receiving data, (b) synchronizing processing between programs, and (c) notifying a partner of errors in the communication.

communication network management (CNM). The process of designing, installing, operating, and managing distribution of information and control among users of communication systems.

communications infrastructure. In the AIX operating system, a framework of communication that consists of a postmaster, an object registration service, a startup file, communication protocols, and application programming interfaces.

Communications Server. An IBM licensed program that supports (a) the development and use of application programs across two or more connected systems or workstations, (b) multiple concurrent connections that use a wide range of protocols, and (c) several application programming interfaces (APIs) that may be called concurrently and that are designed for client/server and distributed application programs. Communications Server includes the necessary interfaces for network management and is available on several operating systems (such as AIX, OS/2® Warp, OS/390, and Windows NT®).

community. In the Simple Network Management Protocol (SNMP), an administrative relationship between entities.

community name. In the Simple Network Management Protocol (SNMP), a string of octets identifying a community.

component description file (CDF). In the context of the Application Management Specification (AMS), an application description file that contains information about a specific component in a management-ready application. Each management-ready application can contain multiple components, each of which is represented by one component description file.

Computing Center Management System (CCMS). The SAP interface for monitoring a SAP R/3 system.

configuration. (1) The manner in which the hardware and software of an information processing system are organized and interconnected. (T) (2) The devices and programs that make up a system, subsystem, or network.

Configuration Application. See MLM Configuration Application.

Configuration Change Management System (CCMS). In a Tivoli environment, a distributed, hierarchical database in which configuration data is stored for use by systems management applications in effecting configuration changes on groups of systems.

configuration file. A file that specifies the characteristics of a system device or network.

configuration management. The control of information necessary to identify both physical and logical information system resources and their relationship to one another.

configuration program. In Tivoli Software Distribution, a feature that enables a Tivoli administrator to perform operations (a) before or after file package distributions, (b) before or after file package removal, (c) during a file package commit operation, or (d) after an error stops a distribution or removal operation.

configuration repository. In a Tivoli environment, the relational database that contains information that is collected or generated by Tivoli applications. Following are examples of the information that is stored in the configuration repository:

- Tivoli Enterprise Console stores information regarding events.
- Tivoli Inventory stores information regarding hardware, software, system configuration, and physical inventory.
- Tivoli Software Distribution stores information regarding file package operations.
connector class. In Tivoli NetView, an object class used for objects that connect different parts of the network and that route or switch traffic between these parts. This class includes gateways, repeaters (including multiport repeaters), and bridges. Contrast with network class.

console event. In a Tivoli environment, an event sent to the Tivoli Enterprise Console.

container. A visual user-interface component that holds objects.

control data set (CDS). In the NetView Performance Monitor (NPM), a System Modification Program (SMP) data set used in the NPM installation process.

control desk. In Tivoli NetView, a component of the graphical user interface (GUI) that enables the network operator to group application program instances together.

control program. (1) A computer program designed to schedule and to supervise the execution of programs of a computer system. (I) (A) (2) The part of the AIX operating system that determines the order in which basic functions should be performed.

control statement. In Tivoli NetView for OS/390, a statement in a command list that controls the processing sequence of the command list or allows the command list to send messages to the operator and receive input from the operator.

COSA. See Common Object Request Broker Architecture.

correlation activity. See event correlation.

CPB. See collection point block.

CPE. See command profile editor.

CPI-C. See Common Programming Interface for Communications.

critical resource. In the NetView Graphic Monitor Facility, a resource that is considered important to the operation of the network and therefore has a high aggregation priority.

cron table. In the AIX operating system, a table that is used to schedule application programs and processes. "Cron" is an abbreviation for "chronological."

cross-domain network manager session. A session between two network managers (for example, Tivoli NetView for OS/390) in separate domains.

cross-system coupling facility (XCF). A component of the MVS™ operating system that provides functions to support cooperation between authorized programs running within a sysplex.

CSCF. See central site control facility.

current directory. See working directory.

Customer Information Control System (CICS). An IBM licensed program that provides online transaction processing services and management for critical business applications. CICS runs on many IBM and non-IBM platforms (from the desktop to the mainframe) and is used in various types of networks that range in size from a few terminals to many thousands of terminals. The CICS application programming interface (API) enables programmers to port applications among the hardware and software platforms on which CICS is available. Each product in the CICS family can interface with the other products in the CICS family, thus enabling interproduct communication.

custom monitor. In Tivoli Distributed Monitoring, a monitor that is implemented as a script or program by the Tivoli administrator.

D

daemon. A program that runs unattended to perform a standard service. Some daemons are triggered automatically to perform their task; others operate periodically.

DASD conservation option. In Tivoli NetView for OS/390, an installation option that allows Tivoli NetView for OS/390 to be installed without the online help facility and hardware monitor data presentation panels.

database. (1) A collection of data with a given structure for accepting, storing, and providing, on demand, data for multiple users. (T) (2) A collection of interrelated data organized according to a database schema to serve one or more applications. (T) (3) A collection of data fundamental to a system. (A) (4) A collection of data fundamental to an enterprise. (A)

data model. (1) A logical view of the organization of data in a database. (T) (2) In a database, the user's logical view of the data in contrast to the physically stored data, or storage structure. (A) (3) A description of the organization of data in a manner that reflects the information structure of an enterprise. (A)

data modeling. A structured set of techniques for defining and recording business information requirements. It is a depiction of the user's view of the data needs of the organization in a consistent and rigorous fashion. The data model eventually serves as the basis for translation to computer system databases.

data services command processor (DSCP). In Tivoli NetView for OS/390, a component that structures a request for recording and retrieving data in the application program's database and for soliciting data from a device in the network.
data services manager (DSM). In Tivoli NetView for OS/390, a function that provides VSAM services for data storage and retrieval.

data services request block (DSRB). In Tivoli NetView for OS/390, the control block that contains information that a data services command processor (DSCP) needs to communicate with the data services task (DST).

data services task (DST). In Tivoli NetView for OS/390, the subtask that gathers, records, and manages data in a VSAM file or a network device that contains network management information.

data type. In Tivoli NetView for OS/390, one of the three elements, which also include display type and resource type, that are used to describe the organization of panels. Data types include alerts, events, and statistics.

dce-pipe-pull. A Printing Systems Manager (PSM) document transfer method in which the client saves documents in a file and transfers the address of the file to the server. The file is later transferred to the server upon request from the server. This is an efficient transfer method for large jobs. Contrast with with-request.

default policy. In a Tivoli environment, a set of resource property values that are assigned to a resource when the resource is created.

definition statement. (1) In VTAM, the statement that describes an element of the network. (2) In NCP, a type of instruction that defines a resource to the NCP.

defragmentation. The process of running a software utility to rewrite fragmented data to contiguous sectors of a computer storage medium to improve access and retrieval time. Contrast with fragmentation.

demand poll. In Tivoli NetView, a polling operation initiated by the user.

deployment management. The Tivoli management discipline that addresses the automation of configuration and change management activities for the ever-evolving components of a network computing system. See availability management, operations and administration, and security management.

desktop. See Tivoli desktop.

Desktop Management Interface (DMI). A protocol-independent set of application programming interfaces (APIs) defined by the Desktop Management Task Force (DMTF). These interfaces give management application programs standardized access to information about hardware and software in a system.

Desktop Management Task Force (DMTF). An alliance of computer vendors that was convened to define streamlined management of the diverse operating systems commonly found in an enterprise.

developer key. In the context of SAP application software, a key that is provided by SAP for a developer’s use in creating or changing Advanced Business Application Programming (ABAP) objects.

DFSMdfp™. A DFSMS/MVS® component and a base element of OS/390 that provides functions for storage management, data management, program management, device management, and distributed data access (“dfp” represents “data facility product”).

DFSMdss™. A DFSMS/MVS component and a base element of OS/390 that is used in copying, moving, dumping, defragmenting, and restoring data sets and volumes (“dss” represents “data set services”).

DFSMSdsm™. A DFSMS/MVS component and a base element of OS/390 that is used in backing up data, in recovering data, in managing storage space on volumes in the storage hierarchy, and in disaster recovery (“dsm” represents “hierarchical storage manager”).

DFSM/MVS. An IBM licensed program that provides storage, data, and device management functions in an MVS/ESA™ Version 5 or an OS/390 environment. DFSM/MVS includes these components: DFSMdfp, DFSMdsss, DFSMSdsm, and DFSMSrm™. “DFSM” represents “Data Facility Storage Management Subsystem.”

DFSMRmm™. A DFSMS/MVS component and base element of OS/390 that manages removable media (“rmm” represents “removable media manager”).

DHCP. See Dynamic Host Configuration Protocol.

directory. In a hierarchical file system, a container for files or other directories. See bath.

discriminator. An object that enables a system to select operations and event reports relating to other managed objects. See event forwarding discriminator.

display type. In Tivoli NetView for OS/390, one of the three elements, which also include data type and resource type, that are used to describe the organization of panels. Display types include total, most recent, user action, and detail.

distributed computing. See network computing.

Distributed Monitoring engine. In a Tivoli environment, the client software that is installed on each managed node, gateway, and endpoint that is being monitored by Tivoli Distributed Monitoring. The Distributed Monitoring engine monitors resources, compares data from monitored resources against configured thresholds, and runs automated responses.

Distributed Monitoring proxy. See endpoint.
distribution program. See configuration program.

DMI. See Desktop Management Interface.

DMTF. See Desktop Management Task Force.

domain. (1) That part of a computer network in which the data processing resources are under common control. (T) (2) See Administrative Domain and domain name.

domain name. In the Internet suite of protocols, a name of a host system. A domain name consists of a sequence of subnames separated by a delimiter character. For example, if the fully qualified domain name (FQDN) of a host system is `ralvm7.vnet.ibm.com`, each of the following is a domain name:

- `ralvm7.vnet.ibm.com`
- `vnet.ibm.com`
- `ibm.com`

double recording. In Tivoli NetView for OS/390, pertaining to the recording of certain individual events under two resource levels.

downcall. In a Tivoli environment, a method invocation from the TMR server or the gateway “down” to an endpoint. Contrast with upcall.

drag and drop. To directly manipulate an object by moving it and placing it somewhere else using a pointing device (such as a mouse).

dynamic encapsulation. (1) In object-oriented programming, the technique that is used to hide the inherent details of an object. This technique is also known as “information hiding.” (2) In object-oriented programming, a software technique in which data is packaged with corresponding procedures. In CORBA, the object is the mechanism for encapsulation.

dynamic Host Configuration Protocol (DHCP). A protocol defined by the Internet Engineering Task Force (IETF) that is used for dynamically assigning IP addresses to computers in a network.

e-business. Either (a) the transaction of business over an electronic medium such as the Internet or (b) any organization (for example, commercial, industrial, nonprofit, educational, or governmental) that transacts its business over an electronic medium such as the Internet. An e-business combines the resources of traditional information systems with the vast reach of an electronic medium such as the Internet (including the World Wide Web, intranets, and extranets); it connects critical business systems directly to critical business constituencies—customers, employees, and suppliers. The key to becoming an e-business is building a transaction-based Web site in which all core business processes (especially all processes that require a dynamic and interactive flow of information) are put online to improve service, cut costs, and sell products.

e-CB. See event control block.

e-commerce. The subset of e-business that involves the exchange of money for goods or services purchased over an electronic medium such as the Internet.

e-FD. See event forwarding discriminato.

EIF. See Tivoli Event Integration Facility.

EMS. See event management services.

encapsulation. (1) In object-oriented programming, the technique that is used to hide the inherent details of an object. This technique is also known as “information hiding.” (2) In object-oriented programming, a software technique in which data is packaged with corresponding procedures. In CORBA, the object is the mechanism for encapsulation.

endpoint. (1) In a Tivoli environment, a Tivoli client that is the ultimate recipient for any type of Tivoli operation. (2) In a Tivoli environment, a Tivoli service that runs on multiple operating systems and performs Tivoli operations on those systems, thereby enabling the Tivoli Management Framework to manage the systems as Tivoli clients.

endpoint list. In a Tivoli environment, a list of all endpoint clients in the Tivoli Management Region with their assigned gateways. See endpoint manager.

endpoint manager. In a Tivoli environment, a service that runs on the Tivoli server, assigns endpoint clients to gateways, and maintains the endpoint list.

endpoint method. In a Tivoli environment, a method that runs on an endpoint client as the result of a request from other managed resources in the Tivoli Management Region. Results of the method are forwarded first to the gateway, then to the calling managed resource.

Enhanced X-Windows Toolkit. (1) In the AIX operating system, a collection of basic functions for developing a variety of application environments. Toolkit functions manage Toolkit initialization, widgets, memory, events, geometry, input focus, selections, resources, translation of events, graphics contexts, pixmaps, and errors. (2) See AIXwindows Toolkit and X Window System.

entity. Any concrete or abstract thing of interest, including associations among things; for example, a person, object, event, or process that is of interest in the context under consideration, and about which data may be stored in a database. (T)

entry point (EP). (1) The address or label of the first instruction executed on entering a computer program,
routine, or subroutine. A computer program, routine, or subroutine may have a number of different entry points, each perhaps corresponding to a different function or purpose. (1) (A) (2) In SNA, a type 2.0, type 2.1, type 4, or type 5 node that provides distributed network management support. It sends network management data about itself and the resources it controls to a focal point for centralized processing, and it receives and executes focal-point initiated commands to manage and control its resources.

**EP.** See *entry point*.

**error record template.** In the AIX operating system, a template that describes the error class, error type, error description, probable causes, recommended actions, and failure data for an error log entry.

**euro.** The monetary unit of the European Monetary Union (EMU) that will be introduced alongside national currencies on the first of January 1999. In May 1998, eleven countries were confirmed for EMU membership beginning the first of January 1999: Austria, Belgium, Finland, France, Germany, Ireland, Italy, Luxembourg, the Netherlands, Portugal, and Spain. On the first of January 2002, euro notes and coins (hard currency) will be put into circulation, and national currencies will be withdrawn, probably over a six-month period.

**EuroReady product.** A product is EuroReady if the product, when used in accordance with its associated documentation, is capable of correctly processing monetary data in the euro denomination, respecting the euro currency formatting conventions (including the euro sign). This assumes that all other products (for example, hardware, software, and firmware) that are used with this product are also EuroReady. IBM hardware products that are EuroReady may or may not have an engraved euro sign key on their keyboards.

**EuroReady solution.** A solution is EuroReady when the solution providers have done the following:

1. Analyzed the euro requirements, including the need to comply with relevant European Community (EC) rules
2. Included the appropriate function according to these requirements
3. Clearly demonstrated this by (a) detailing the euro-related requirements, (b) describing how these requirements will be implemented, and (c) declaring when the implementation will be generally available.

**event.** (1) An occurrence of significance to a task (such as the opening of a window or the completion of an asynchronous operation). (2) In the Tivoli environment, any significant change in the state of a system resource, network resource, or network application. An event can be generated for a problem, for the resolution of a problem, or for the successful completion of a task. Examples of events are: the normal starting and stopping of a process, the abnormal termination of a process, and the malfunctioning of a server. (3) See *event report*.

**event adapter.** In a Tivoli environment, software that converts events into a format that the Tivoli Enterprise Console can use and forwards the events to the event server. Using the Tivoli Event Integration Facility, an organization can develop its own event adapters, tailored to its network environment and specific needs.

**event/automation service.** In Tivoli NetView for OS/390, a facility that translates alerts and messages into events for the Tivoli Enterprise Console and translates these events into NetView alerts. The event/automation service communicates with Tivoli NetView for OS/390 using the program-to-program interface (PPI), and it communicates with the Tivoli Enterprise Console using TCP/IP.

**event card.** In Tivoli NetView, a graphical representation, resembling a card, of the information contained in an event sent by an agent to a manager reflecting a change in the status of one of the agent’s managed nodes.

**event class.** In the Tivoli Enterprise Console, a classification for an event that indicates the type of information that the event adapter will send to the event server.

**event console.** In the Tivoli Enterprise Console, a graphical user interface (GUI) that enables system administrators to view and respond to dispatched events from the event server. The Tivoli Event Integration Facility does not directly use or affect event consoles.

**event control block (ECB).** A control block used to represent the status of an event.

**event correlation.** In the Tivoli Enterprise Console, the process of correlating separate events to a common cause. For example, the Tivoli Enterprise Console may receive several *NFS server not responding* events from several different applications, as well as a host down event for the NFS server. The Tivoli Enterprise Console can then correlate the various *NFS server not responding* events to their common cause, which is: the NFS server is “down.” See *rule*.

**event filter.** (1) In a Tivoli environment, software that determines which events are forwarded to a specified destination. Filtering events helps to reduce network traffic. Tivoli administrators configure the event filters. (2) In Tivoli NetView, a logical expression of criteria that determine which events are forwarded to the application program that registers the event filter with the event sieve agent. A filter is referred to as “simple” or “compound” depending on how it is handled by the filter editor.

**event forwarding discriminator (EFD).** A managed object that describes the criteria used to select which event reports are sent and to whom they are sent.
**event group.** In the Tivoli Enterprise Console, a set of events that meet certain criteria. Each event group is represented by an icon on the event console. Tivoli administrators can monitor event groups that are relevant to their specific areas of responsibility.

**event handler.** A collection and correlation point for events and messages.

**event management services (EMS).** In Tivoli NetView, a centralized method of generating, receiving, routing, and logging network events.

**event manager.** In the NetView Graphic Monitor Facility, the component of the host subsystem that receives alert and resolution major vectors from Tivoli NetView for OS/390, translates these major vectors into generic event records, and applies the event status to the resource defined in the Resource Object Data Manager (RODM) cache.

**event report.** The unsolicited report that an event has occurred. In an Open Systems Interconnection (OSI) context, when a managed object emits a notification, the agent uses one or more event forwarding discriminators (EFDs) to find the destinations to which the report is sent.

**event repository.** See configuration repository.

**event server.** In the Tivoli Enterprise Console, a central server that processes events. The event server creates an entry for each incoming event and evaluates the event against a rule base to determine whether it can respond to or modify the event automatically. The event server also updates the event consoles with the current event information. If the primary event server is not available, events can be sent to a secondary event server.

**event sieve.** In Tivoli NetView, an object that is managed by the "ovesmd" daemon, which is the event sieve agent. The event sieve agent stores information about the event sieve object in a database and reads that information when the agent is started. See [event filter](#) and [event forwarding discriminator](#).

**event slot.** In a Tivoli environment, a discrete area (a field) of an event record that contains a specific type of information about an event.

**event specifier.** In the Tivoli Enterprise Console, a rules-language program construct that is used to look for events in the event cache. For example, it can look for duplicate events, an event that matches a user-specified attribute, or an event that occurs within a certain time period. An event specifier is used in building rules, and it dictates how the Tivoli Enterprise Console will handle an event that it receives.

**exception.** An abnormal condition such as an I/O error encountered in processing a data set or a file.

**exclusive set.** In Remote Operations Service (ROPS), an option that indicates whether only the commands in the command list can be processed by ROPS or none of the commands in the command list can be processed by ROPS.

**exclusive submap.** In Tivoli NetView, a submap that is created by an application program wanting the exclusive right to control what happens in the application plane of the submap. Contrast with [shared submap](#).

**exec.** (1) In the AIX operating system, to overlay the current process with another executable program. (2) See fork.

**executable symbol.** In Tivoli NetView, a symbol defined such that double-clicking on it or dragging and dropping it displays the child submap of the parent object that the symbol represents. Contrast with [explodable symbol](#).

**execution target.** In a Tivoli environment, a managed node on which a job or other activity is performed. For example, if an application is being installed on a particular server, that server is the execution target for the installation activity.

**explicit command.** In Tivoli NetView for OS/390, a command that is used to request the display of information that the user would otherwise obtain by navigating through a hierarchy of panels.

**explodable symbol.** In Tivoli NetView, a symbol defined such that double-clicking on it or dragging and dropping it displays the child submap of the parent object that the symbol represents. Contrast with [executable symbol](#).

**export/import.** In Tivoli Software Distribution, a feature that enables a Tivoli administrator to save (export) a file package definition as a text file, to edit the keywords and lists in the definition, and to retrieve (import) the definition from the text file to set the properties for the file package.

**extended enterprise.** The customers, suppliers, distributors, and other business partners with whom a company conducts e-business.

**extranet.** A private, virtual network that uses access control and security features to restrict the usage of one or more intranets attached to the Internet to selected subscribers (such as personnel from a sponsoring company and its business partners).

**failover system.** In Tivoli Manager for R/3, a computer that serves as a transparent backup to a primary computer. The primary computer and the failover
system share access to a common R/3 database, thereby enabling either machine to provide full database support.

**fanout.** In communication, the process of creating copies of a distribution to be delivered locally or to be sent through the network.

**field.** (1) An identifiable area in a window. Examples of fields are: an entry field, into which a user can type or place text, and a field of radio button choices, from which a user can select one choice. (2) The smallest identifiable part of a record. (3) In Tivoli NetView, the building block of which objects are composed. A field is characterized by a field name, a data type (integer, Boolean, character string, or enumerated value), and a set of flags that describe how the field is treated by Tivoli NetView. A field can contain data only when it is associated with an object.

**field registration file (FRF).** In Tivoli NetView, a file used to define fields for use in the object database.

**file name substitution.** In the AIX operating system, the process in which the shell substitutes an alphabetically sorted list of file names in the place of a pattern. The shell recognizes a pattern (as opposed to an alphabetically sorted list of file names in the place of a pattern) by the occurrence of a word (character string) pattern. The shell recognizes a pattern (as opposed to an alphabetically sorted list of file names in the place of a pattern) by the occurrence of a word (character string) pattern. The shell substitutes a file name substitution.

**forefront process.** (1) In the AIX operating system, a process that must run to completion before another command is issued to the shell. The foreground process is in the foreground process group, which is the group that receives the signals generated by a terminal. (2) Contrast with **background process**.

**foreground task.** The task with which the user is interacting. Contrast with **background task**.

**foreign host.** See [remote host](#).

**fork**. In the UNIX operating system, to create and start a child process.

**fpblock.** See [file package block](#).

**FQDN**. See [fully qualified domain name](#).

**fragmentation.** An operating system’s process of writing different parts of a file to contiguous sectors on a computer storage medium when contiguous space that is large enough to contain the entire file is not available. When data is thus fragmented, the time that it takes to access the data may increase because the operating system must search different tracks for information that should be in one location. Contrast with [defragmentation](#).

**FRF.** See [field registration file](#).

**full pathname.** See [absolute path](#).

**fully qualified domain name (FQDN).** In the Internet suite of protocols, the name of a host system that includes all of the subnames of the domain name. An example of a fully qualified domain name is `ravm7.vnet.ibm.com`. See [host name](#).

G

**gadget.** In the AIXwindows Toolkit, a windowless graphical object that looks like its equivalent like-named widget but does not support the translations, actions, or pop-up widget children supplied by that widget.

**gateway.** (1) A functional unit that interconnects two computer networks with different network architectures. A gateway connects networks or systems of different architectures. A bridge interconnects networks or systems with the same or similar architectures. (T) (2) A functional unit that connects two networks or subnetworks having different characteristics, such as different protocols or different policies concerning security or transmission priority. (3) The combination of machines and programs that provide address...
translation, name translation, and system services control point (SSCP) rerouting between independent SNA networks to allow those networks to communicate. A gateway consists of one gateway NCP and at least one gateway VTAM. (4) In a Tivoli environment, software running on a managed node that provides all communication services between a group of endpoints and the rest of the Tivoli environment. This gateway includes the multiplexed distribution (MDist) function, enabling it to act as the fanout point for distributions to many endpoints. (5) See router.

gateway-capable host. A host node that has a defined NETID and SSCPNAME but does not perform gateway control functions, such as cross-network session initiation and termination.

gateway host. (1) A host node that contains a gateway system services control point (SSCP). See gateway-capable host. (2) In the AIX operating system, a host that connects independent networks. It has multiple interfaces, each with a different name and address.

gateway method. In a Tivoli environment, a method that runs on the gateway’s proxy managed node on behalf of the endpoint. Results of the method are forwarded to the calling managed resource.

gcs. See graphic communication server.

gddm®. See Graphical Data Display Manager.

gddm interface for x window system (gddmxd). A graphical interface that formats and displays characters, graphics, and images on workstation display devices that support the X Window System.

gdmmxd. See GDDM interface for X Window System.

gdf. See global description file.

gds. See graphic data server.

gem. See Tivoli Global Enterprise Manager.

general topology manager (GTM). In TivoliNetView, the component that accepts information about resources that are accessed through protocols other than the Internet Protocol (IP), stores this information in a database, and displays it to the user.

generic alert. In SNA management services (SNA/MS), alert information that is encoded using a method in which code points provide an index into short units of stored text. The use of generic alerts prevents the receiver from having to recognize and understand each unique problem for which an alert is sent. Contrast with non-generic alert.

generic collection. In a Tivoli environment, a collection that contains objects representing resources of any type.

gid. See Group ID.

gif. See graphical interchange format.

global description file (GDF). In the context of the Application Management Specification (AMS), an application description file that provides global information about an application such as the application name, the version identifier, and a free-form description of the application. Each version of a management-ready application is represented by one global description file.

GMFHS. See Graphic Monitor Facility host subsystem.

Graphical Data Display Manager (GDDM). In the NetView Performance Monitor (NPM), an IBM licensed program used in conjunction with the Presentation Graphics Feature (PGF) to generate online graphs in the NPM Graphic Subsystem.

graphical interchange format (GIF). A digital format that is used to compress and transfer graphical information over computer networks. For example, GIF is a common format for graphical information on the Internet.

graphical user interface (GUI). A type of computer interface consisting of a visual metaphor of a real-world scene, often of a desktop. Within that scene are icons, representing actual objects, that the user can access and manipulate with a pointing device. Contrast with Command line interface.

graphic communication server (GCS). The part of the NetView Graphic Monitor Facility that manages LU 6.2 sessions used for data transport between (a) Tivoli NetView for OS/390 and the server workstation and (b) the server workstation and its client workstations.

graphic data server (GDS). The part of the NetView Graphic Monitor Facility that receives network management data from Tivoli NetView for OS/390, maintains this data (except for dynamically created view information), and correlates this data with views.

graphic monitor. The graphical user interface (GUI) component of the NetView Graphic Monitor Facility.

Graphic Monitor Facility host subsystem (GMFHS). In Tivoli NetView for OS/390, a component that manages updates to the configuration and status of resources displayed in NetView Graphic Monitor Facility (NGMF) views.

graphics context (GC, Gcontext). In the Enhanced X-Windows Toolkit, the storage area for various kinds of graphics output, such as foreground pixels, background pixels, line widths, and clipping regions. A graphics context can be used only with drawables that have the same root and the same depth as the graphics context.

graphics data file (GDF). A picture definition in a coded format that is used internally by the Graphical
Data Display Manager (GDDM) and, optionally, provides the user with a lower level program interface than the GDDM application programming interface (API).

**group ID (GID).** In the AIX operating system, a number that corresponds to a specific group name. The group ID can often be substituted in commands that take a group name as a value.

**group profile.** In Tivoli User Administration, a profile that a Tivoli administrator uses to define and modify information about a group of users.

**GTM.** See [general topology manager](#).

**GUI.** See [graphical user interface](#).

**H**

**hardcoded.** Pertaining to software instructions that are statically encoded and not intended to be altered.

**hardcopy task (HCT).** In Tivoli NetView for OS/390, the subtask that controls the passage of data between the NetView program and the hardcopy device.

**hardware monitor.** In Tivoli NetView for OS/390, the component that helps identify and solve problems related to physical network elements (as opposed to logical sessions, which are managed by the session monitor). Contrast with [session monitor](#).

**HCT.** See [hardcopy task](#).

**heartbeat.** In software products, a signal that one entity sends to another to convey that it is still active.

**home submap.** In Tivoli NetView, the first submap that appears when a map is opened. Each map has a home submap. When new maps are created, the home submap is the root submap.

**hook.** A location in a computer program where an instruction is inserted for invoking a particular function.

**host.** (1) A computer that is connected to a network (such as the Internet or an SNA network) and provides an access point to that network. Also, depending on the environment, the host may provide centralized control of the network. The host can be a client, a server, or both a client and a server simultaneously. (2) In a Tivoli environment, a computer that serves as a managed node for a profile distribution. (3) See [host processor](#).

**host name.** In the Internet suite of protocols, the name given to a machine. Sometimes, "host name" is used to mean [fully qualified domain name](#) or a name that is used to mean the most specific subname of a fully qualified domain name. For example, if ralvm7.vnet.ibm.com is the fully qualified domain name, either of the following may be considered the host name:

- ralvm7.vnet.ibm.com
- ralvm7

**host namespace profile.** In Tivoli Enterprise software, a profile that contains information about the list of hosts and their properties, such as host IP addresses and host aliases.

**host processor.** (1) A processor that controls all or part of a user application network. (2) In a network, the processing unit in which the data communication access method resides.

**host transit time.** In the NetView Performance Monitor (NPM), the average time (in seconds) that all transactions spend in the host. It includes both VTAM and application time. It is also reported as an average for the transactions originating at the logical unit for which data collection is occurring.

**HTML.** See [Hypertext Markup Language](#).

**HTTP.** See [Hypertext Transfer Protocol](#).

**hub.** In a network, a point at which circuits are either connected or switched. For example, in a star network, the hub is the central node; in a star/ring network, it is the location of wiring concentrators.

**Hypertext Markup Language (HTML).** A markup language that is specified by an SGML document type definition (DTD) and is understood by all Web servers.

**Hypertext Transfer Protocol (HTTP).** In the Internet suite of protocols, the protocol that is used to transfer and display hypertext documents.

**I**

**IAB.** See [Internet Architecture Board](#).

**ICMP.** See [Internet Control Message Protocol](#).

**IDL.** See [Interface Definition Language](#).

**IETF.** See [Internet Engineering Task Force](#).

**immediate command.** In Tivoli NetView for OS/390, a command (such as GO, RESET, or LOGOFF) that begins processing as soon as the operator enters it, possibly preempting other ongoing processing. All other commands are called "regular commands" and are processed by a "regular command processor." Regular commands can run concurrently with other regular commands and can be interrupted by immediate commands. Most commands and all command lists are regular commands.

**IMS™.** See [Internet Management Specification](#).

**indicator.** In Tivoli Distributed Monitoring, an icon on the Tivoli desktop that graphically displays the status of a monitor that has been associated with it. The icon
resembles a thermometer, which the Tivoli administrator can read to determine the status of the monitor.

**indicator collection.** In a Tivoli environment, a single location from which a Tivoli administrator can determine the status of monitors in different profiles, as well as clear and reset alarmed states.

**instance.** In object-oriented programming, an object created by instantiating a class.

**instantiate.** In object-oriented programming, to represent a class abstraction with a concrete instance of the class.

**instrument.** In application or system software, to use monitoring functions to provide performance and other information to a management system.

**instrumentation.** In application or system software, either (a) monitoring functions that provide performance and other information to a management system or (b) the use of monitoring functions to provide performance and other information to a management system.

**intelligent agent.** Software that monitors conditions or actions on a network node and contains logic enabling it to respond to these conditions or actions.

**interactive chart utility (ICU).** A utility provided by the Graphical Data Display Manager (GDDM) to allow basic graphic handling capability and a menu-driven generation of different forms of graphs. ICU is a part of the presentation graphics feature.

**Interface Definition Language (IDL).** In CORBA, a declarative language that is used to describe object interfaces, without regard to object implementation.

**Internet Architecture Board (IAB).** The technical body that oversees (at a high level) the work of the Internet Engineering Task Force (IETF). The IAB approves the membership of the IETF.

**Internet Control Message Protocol (ICMP).** The protocol used to handle errors and control messages in the Internet Protocol (IP) layer. Reports of problems and incorrect datagram destinations are returned to the original datagram source.

**Internet Engineering Task Force (IETF).** The task force of the Internet Architecture Board (IAB) that is responsible for solving the short-term engineering needs of the Internet. The IETF consists of numerous working groups, each focused on a particular problem. Internet standards are typically developed or reviewed by individual working groups before they can become standards.

**Internet Management Specification (IMS).** A draft specification for an open standard for managing Internet resources and services.

**internet object.** In Tivoli NetView, a node or a network that can be accessed by the Internet Protocol (IP).

**Internet Protocol (IP).** In the Internet suite of protocols, a connectionless protocol that routes data through a network or interconnected networks. IP acts as an intermediary between the higher protocol layers and the physical network. However, this protocol does not provide error recovery and flow control and does not guarantee the reliability of the physical network.

**Internet service provider (ISP).** An organization that provides access to the Internet.

**Internetwork Packet Exchange (IPX).** The network protocol used to connect Novell's servers, or any workstation or router that implements IPX, with other workstations. Although similar to the Internet Protocol (IP), IPX uses different packet formats and terminology.

**interprocess communication (IPC).** The process by which programs communicate data to each other and synchronize their activities. Semaphores, signals, and internal message queues are common methods of interprocess communication.

**intranet.** A private network that integrates Internet standards and applications (such as Web browsers) with an organization's existing computer networking infrastructure.

**IP.** See [Internet Protocol](#).

**IPC.** See [interprocess communication](#).

**IPX.** See [Internetwork Packet Exchange](#).

**ISP.** See [Internet service provider](#).

**IT.** Information technology.

**J**

**Java.** An object-oriented programming language for portable interpretive code that supports interaction among remote objects. Java was developed and specified by Sun Microsystems, Incorporated.

**JavaBeans.** A platform-independent, software component technology for building reusable Java components called "beans." Once built, these beans can be made available for use by other software engineers or can be used in Java applications. Also, using JavaBeans, software engineers can manipulate and assemble beans in a graphical drag-and-drop development environment.

**Java Database Connectivity (JDBC).** An application programming interface (API) that has the same characteristics as Open Database Connectivity (ODBC) but is specifically designed for use by Java database applications. Also, for databases that do not have a JDBC driver, JDBC includes a JDBC to ODBC bridge.
which is a mechanism for converting JDBC to ODBC; it presents the JDBC API to Java database applications and converts this to ODBC. JDBC was developed by Sun Microsystems, Inc. and various partners and vendors.

Java Management Application Programming Interface (JMAPI). A specification proposed by Sun Microsystems that defines a core set of application programming interfaces for developing tightly integrated system, network, and service management applications. The application programming interfaces could be used in diverse computing environments that encompass many operating systems, architectures, and network protocols.

JDBC. See Java Database Connectivity.

JMAPI. See Java Management Application Programming Interface.

job. (1) A unit of work defined by a user that is to be accomplished by a computer. Loosely, the term job is sometimes used to refer to a representation of a job. This representation may include a set of computer programs, files, and control statements to the operating system. (I) (A) (2) A Printing Systems Manager (PSM) object that represents a request to print one or more documents in a single printing session. (3) In a Tivoli environment, a resource consisting of a task and its preconfigured execution parameters. Among other things, the execution parameters specify the set of hosts on which the job is to execute.

JPEG. A standard format for storing compressed true-color images. “JPEG” represents “Joint Photographic Experts Group,” which is the name of the committee that developed this standard format.

key. In computer security, a sequence of symbols that is used with a cryptographic algorithm for encrypting or decrypting data. See private key and public key.

keyword. (1) In programming languages, a lexical unit that, in certain contexts, characterizes some language construct; for example, in some contexts, IF characterizes an if-statement. A keyword normally has the form of an identifier. (I) (2) One of the predefined words of an artificial language. (A) (3) A name or symbol that identifies a parameter. (4) The part of a command operand that consists of a specific character string (such as DNAME= ). (5) See keyword operand.

keyword operand. (1) An operand that consists of a keyword followed by one or more values (such as DNAME=HELLO ). (2) Contrast with positional operand. (3) See definition statement.

keyword parameter. A parameter that consists of a keyword followed by one or more values.

L

LAN. See local area network.

LAN Network Manager (LNM). An IBM licensed program that enables a user to manage and monitor LAN resources from a central workstation.

LCCM. See link connection component manager.

LCSM. See link connection subsystem manager.

link connection component manager (LCCM). The transaction program that manages the configuration of the link connection.

link connection subsystem manager (LCSM). The transaction program that manages the sequence of link connection components that belong to a link connection.

Link Problem Determination Aid (LPDA®). A series of procedures that are used to test the status of and to control DCEs, the communication line, and the remote device interface. These procedures, or a subset of them, are implemented by host programs (such as Tivoli NetView for OS/390 and VTAM), communication controller programs (such as NCP), and IBM LPDA DCEs. See LPDA-1 and LPDA-2.

LNM. See LAN Network Manager.

local area network (LAN). (1) A computer network located on a user’s premises within a limited geographical area. Communication within a local area network is not subject to external regulations; however, communication across the LAN boundary may be subject to some form of regulation. (T) (2) A network in which a set of devices are connected to one another for communication and that can be connected to a larger network.
local distribution. In a Tivoli environment, a distribution to target machines in the same Tivoli Management Region as the source machine.

local overrides. In a Tivoli environment, a feature of all profile-based Tivoli applications—except for Tivoli Software Distribution—that allows changes made at the endpoint profile to override those in a distributed profile.

local registration file (LRF). In Tivoli NetView, a file that provides information about an agent or daemon, such as the name, the location of the executable code, the names of processes dependent on the agent or daemon, and details about the objects that an agent manages.

local topology database. A database in an APPN or LEN node containing an entry for each transmission group (TG) having at least one end node for an endpoint. In an end node, the database has one entry for each TG connecting to the node. In a network node, the database has an entry for each TG connecting the network node to an end node. Each entry describes the current characteristics of the TG that it represents. A network node has both a local and a network topology database while an end node has only a local topology database.

lock. The means by which integrity of data is ensured by preventing more than one user from accessing or changing the same data or object at the same time.

logged-on operator. (1) In Tivoli NetView for OS/390, an operator station task that requires a terminal and a logged-on user. (2) Contrast with autotask.

LPDA. See Link Problem Determination Aid.

LPDA-1. The first version of the Link Problem Determination Aid (LPDA) command set. LPDA-1 is not compatible with LPDA-2.

LPDA-2. The second version of the Link Problem Determination Aid (LPDA) command set. LPDA-2 provides all of the functions of LPDA-1; it also supports commands such as the following:
- DCE configuration
- Dial
- Set transmit speed
- Commands to operate a contact that can control external devices.

LRF. See local registration file.

LUC session. Communication, using LU type 0 protocols, between the LUC tasks of two Tivoli NetView for OS/390 programs. This communication is similar to an LU 6.2 conversation.

LUC task. A Tivoli NetView for OS/390 task, denoted by the NetView domain ID concatenated with the literal “LUC” (for example, CNM01LUC), that serves as the endpoint of an LUC session.

LU group. (1) In the NetView Performance Monitor (NPM), a file containing a list of related or unrelated logical units. The LU group is used to help simplify data collection and analysis. (2) In Tivoli NetView for OS/390, a grouping of logical units according to some affinity, such as their link to the same VTAM generic resource or VTAM USERVAR.

LU 6.2 verb. A syntactical unit in the LU 6.2 application programming interface representing an operation.

M

macroinstruction. (1) An instruction in a source language that is to be replaced by a defined sequence of instructions in the same source language and that may also specify values for parameters in the replaced instructions. (T) (2) In assembler programming, an assembler language statement that causes the assembler to process a predefined set of statements called a macro definition. The statements normally produced from the macro definition replace the macroinstruction in the program.

managed node. (1) In Internet communications, a workstation, server, or router that contains a network management agent. In the Internet Protocol (IP), the managed node usually contains a Simple Network Management Protocol (SNMP) agent. (2) In a Tivoli environment, any managed resource on which the Tivoli Management Framework is installed.

managed object. (1) A component of a system that can be managed by a management application. (2) The systems management view of a resource that can be managed through the use of systems management protocols.

managed resource. In a Tivoli environment, any hardware or software entity (machine, service, system, or facility) that is represented by a database object and an icon on the Tivoli desktop. Managed resources must be a supported resource type in a policy region and are subject to a set of rules. Managed resources include, but are not limited to, managed nodes, task libraries, monitors, profiles, and bulletin boards.

management by subscription. In a Tivoli environment, the concept of managing network resources by creating sets of profiles and distributing the profiles (through profile managers) to physical entities (Tivoli resources), called subscribers.

Management Information Base (MIB). (1) A collection of objects that can be accessed by means of a network management protocol. (2) A definition for management information that specifies the information available from a host or gateway and the operations allowed. (3) In
OSI, the conceptual repository of management information within an open system. (4) See MIB module.

Management Information Format (MIF). The Desktop Management Interface (DMI) specification that defines the syntax for describing management information about the hardware and software components that can be installed on a computer system.

management module. In a Tivoli environment, a file that contains the management information and instrumentation for enabling a particular application or business system to be managed by Tivoli management software. This file may be in the form of a Tivoli install image or an application management package. Types of management modules include base modules, Tivoli GEM modules, and Tivoli Plus modules. See Tivoli Module Builder and Tivoli Module Designer.

management region. In Tivoli NetView, the set of managed objects on a particular map that defines the extent of the network that is being actively managed. The management region may vary across maps.

management services (MS). (1) One of the types of network services in control points (CPs) and physical units (PUs). Management services are the services provided to assist in the management of SNA networks, such as problem management, performance and accounting management, configuration management, and change management. (2) Services that assist in the management of systems and networks in areas such as problem management, performance management, business management, operations management, configuration management, and change management.

management services focal point (MSFP). For any given management services discipline (for example, problem determination or response time monitoring), the control point that is responsible for that type of network management data for a sphere of control. This responsibility may include collecting, storing, or displaying the data, or all of these. (For example, a problem determination focal point is a control point that collects, and that may store or display, problem determination data.)

manager. (1) In systems management, a user that, for a particular interaction, has assumed a manager role. (2) An entity that monitors or controls one or more managed objects by (a) receiving notifications regarding the objects and (b) requesting management operations to modify or query the objects. (3) A system that assumes a manager role.

manager role. In systems management, a role assumed by a user where the user is capable of issuing management operations and of receiving notifications.

man page. In UNIX systems, one page of online documentation. “Man page” is an abbreviation for “manual page.” Each UNIX command, utility, and library function has an associated man page that can be viewed by entering this command: man command name.

map. In Tivoli NetView, a database represented by a set of related submaps that provide a graphical and hierarchical presentation of a network and its systems.

mapper. In Tivoli NetView for OS/390, a function that records errors from resources attached to a communication controller or from certain channel-attached devices.

marshall. To copy data into a remote procedure call (RPC) packet. Stubs perform marshalling. Contrast with unmarshall.

MCSL. See Monitoring Collection Specification Language.

MDist. Multiplexed distribution. In a Tivoli environment, a service that enables efficient distribution of large amounts of data across complex networks.

menu bar. (1) The area near the top of a window, below the title bar and above the rest of the window, that contains choices that provide access to other menus. (2) In the AIX operating system, a rectangular area at the top of the client area of a window that contains the titles of the standard pull-down menus for that application.

message style. In Tivoli Distributed Monitoring, the amount and format of information presented by certain monitors.

method. (1) In object-oriented design or programming, the software that implements the behavior specified by an operation. (2) In Tivoli NetView for OS/390, a program that runs in the Resource Object Data Manager (RODM) address space and communicates with RODM using an application programming interface (API). Methods are usually small programs that perform specific tasks on data in the data cache.

MIB. See Management Information Base.

MIB application program. A systems management application program used to monitor network devices.

MIB module. In the Simple Network Management Protocol (SNMP), a collection of objects relating to a common management area. See MIB variable.

MIB object. See MIB variable.

MIB tree. In the Simple Network Management Protocol (SNMP), the structure of the Management Information Base (MIB).

MIB variable. In the Simple Network Management Protocol (SNMP), a specific instance of data defined in a MIB module.
**MIB view.** In the Simple Network Management Protocol (SNMP), the collection of managed objects, known to the agent, that is visible to a particular community.

**MIB walking.** In the Simple Network Management Protocol (SNMP), a technique of looking for Management Information Base (MIB) tree information when it is presented in a hierarchical format.

**Mid-Level Manager (MLM).** In Tivoli NetView, the component that performs certain systems and network management tasks (for example, polling, status monitoring, and node discovering) for a defined set of Simple Network Management Protocol (SNMP) devices in the network, thereby offloading these tasks from Tivoli NetView.

**MIF.** See Management Information Format.

**MIPS.** A measure of computer processing performance that is equal to one million instructions per second.

**MLM.** See Mid-Level Manager.

**MLM Configuration Application.** A Tivoli NetView feature that is used to configure the Mid-Level Manager (MLM).

**MNPS.** See multinode persistent session.

**module.** See management module.

**monitor.** (1) A device that observes and records selected activities within a data processing system for analysis. Possible uses are to indicate significant departure from the norm, or to determine levels of utilization of particular functional units. (T) (2) Software or hardware that observes, supervises, controls, or verifies operations of a system. (A) (3) Software that monitors specific applications or the systems on which the applications rely. Monitors typically monitor information such as available disk space or application errors and compare the information to defined thresholds. When thresholds are exceeded, either system or network administrators can be notified, or an automated response can be performed. (4) In the NetView Graphic Monitor Facility, to open a view that can receive status changes from Tivoli NetView for OS/390. Problem determination and correction can be performed directly from the view. Contrast with browse.

**monitoring collection.** In Tivoli Distributed Monitoring, a collection of predefined monitors. Several monitoring collections are packaged with Tivoli Distributed Monitoring, but Tivoli administrators can use custom-developed and third-party monitoring collections as well. See Custom monitors.

**Monitoring Collection Specification Language (MCSL).** A proprietary programming language that is owned by Tivoli Systems Inc. and is used to define monitoring collections for Tivoli Distributed Monitoring.

**MPM.** See MultiPlatform Manager.

**MS.** See management services.

**MSFP.** See management services focal point.

**multinode persistent session (MNPS).** An LU-LU session that is retained after the failure of VTAM, the operating system, or the hardware.

**MultiPlatform Manager (MPM).** An application programming interface (API) that was developed by a group of leading technology vendors, including Tivoli Systems Inc., and that enables disparate management systems to interoperate with each other. Tivoli LAN Access and Tivoli IT Director support this API, which means that Tivoli Enterprise software or Tivoli IT Director can provide IT managers with unifying, centralized control over disconnected management resources.

**Multiple Virtual Storage/Operator Communication Control Facility (MVS/OCCF).** A facility that intercepts messages from the MVS supervisor. Tivoli NetView for OS/390 and MVS/OCCF help a network operator control multiple MVS systems from a central site.

**multiplexed distribution (MDist).** See MDist.

**MultiSystem Manager.** In Tivoli NetView for OS/390, the component that manages non-SNA resources, such as those in IP networks, NetWare networks, LAN Network Manager networks, and LAN NetView Management Utilities networks.

**multitiered application.** An application that is deployed on more than one physical machine. A client/server application is a common multitiered application in which there are two tiers: the client tier (for example, the presentation and the graphical user interface) and the server tier (for example, the service and the database).

**MVS/OCCF.** See Multiple Virtual Storage/Operator Communication Control Facility.

**MVS system symbol.** In a sysplex where a customer runs a copy of a given program (such as CICS or Tivoli NetView for OS/390) on more than one MVS image, a symbol that the customer can use to write generic JCL for use by each instance of the given program. An MVS system symbol behaves like a program variable that the sysplex resolves at execution time with the value that is appropriate to the MVS image on which the program instance is running.

**name registry.** In a Tivoli environment, a name service consisting of a two-dimensional table that maps
resource names to resource identifiers and corresponding information within a Tivoli Management Region.

**name translation.** In SNA network interconnection, the conversion of logical unit names, logon mode table names, and class-of-service names used in one network to equivalent names for use in another network. This function can be provided through Tivoli NetView for OS/390 and invoked by a gateway system services control point (SSCP) when necessary. See [alias name](#).

**NAT.** See [network address translation](#).

**navigate.** In the NetView Graphic Monitor Facility, to move between levels in the view hierarchy.

**navigation tree.** In Tivoli NetView, a component of the graphical user interface (GUI) that displays a hierarchy of open submaps illustrating the parent-child relationship. The navigation tree enables the network operator to determine which submaps are currently open and to close, restore, or raise the windows that contain submaps.

**NCCF.** In Tivoli NetView for OS/390, a command that starts the NetView command facility. Also, the use of the abbreviation “NCCF” indicates that various panels and functions are part of the command facility.

**nested file package.** In Tivoli Software Distribution, a file package that is added as an entry to another file package.

**NetBIOS.** (1) Network Basic Input/Output System. A standard interface to networks, IBM personal computers (PCs), and compatible PCs, that is used on LANs to provide message, print-server, and file-server functions. Application programs that use NetBIOS do not need to handle the details of LAN data link control (DLC) protocols. (2) See [Basic Input/Output System](#).

**NetView.** See [Tivoli NetView](#) and [Tivoli NetView for OS/390](#).

**NetView AutoBridge.** In Tivoli Service Desk for OS/390, an application program that works with the NetView Bridge Adapter to update the Tivoli Service Desk for OS/390 database and to automate network monitoring. The NetView AutoBridge receives data from NetView alerts, messages, and other applications and uses this data to build and perform Tivoli Service Desk for OS/390 transactions.

**NetView Bridge.** In Tivoli NetView for OS/390, a set of application programming interfaces (APIs) that enable Tivoli NetView for OS/390 to interact with various types of databases in the OS/390 environment.

**NetView Bridge Adapter.** In Tivoli Service Desk for OS/390, a feature that provides a connection between the NetView Bridge and the Tivoli Service Desk for OS/390 database. The NetView Bridge Adapter enables the Tivoli Service Desk for OS/390 to act as a NetView database server and works with the NetView AutoBridge or other NetView applications to access problem records logged in the Tivoli Service Desk for OS/390 database.

**NetView command authorization table.** In Tivoli NetView for OS/390, a set of entries that define an operator's authorization for accessing commands and (depending on the level of granularity that an enterprise chooses) command keywords and keyword values.

**NetView command list language.** In Tivoli NetView for OS/390, an interpretive language that is unique to the NetView program and that is used to write NetView command lists in environments where REXX is not supported.

**NetView Graphic Monitor Facility (NGMF).** In Tivoli NetView for OS/390, a function that provides the network operator with a graphical topological representation of a network and allows the operator to manage the network interactively.

**NetView help desk.** In Tivoli NetView for OS/390, an online information facility that guides the help desk operator through problem management procedures.

**NetView Installation and Administration Facility/2 (NIAF/2).** An OS/2-based tool that allows new users of Tivoli NetView for OS/390 or users migrating from a prior release to install, administer, and maintain Tivoli NetView for OS/390. NIAF/2 replaces the Interactive System Productivity Facility-based (ISPF-based) NetView Installation Facility.

**NetView management console.** See [topology console](#).

**NetView management console server.** See [topology server](#).

**NetView-NetView task (NNT).** In Tivoli NetView for OS/390, the task under which a cross-domain NetView operator session runs. See [operator station task](#).

**NetView Performance Monitor (NPM).** An IBM licensed program that collects, monitors, analyzes, and displays data relevant to the performance of a VTAM telecommunication network. It runs as an online VTAM application program.

**NetWare managed site.** In a Tivoli environment, a resource that represents (a) a Novell NetWare server on which the Tivoli NetWare repeater (TNWR) is installed and (b) one or more clients. A NetWare managed site enables profiles to be distributed through the NetWare server to one or more specified client PCs using either TCP/IP or IPX.

**network address translation (NAT).** In a firewall, the conversion of secure IP addresses to external
registered addresses. This enables communication with external networks but masks the IP addresses that are used inside the firewall.

**network class.** In Tivoli NetView, an object class used for symbols that represent compound objects that may contain objects such as hosts and network devices. Contrast with connector class.

**network computing.** The use of a scalable distributed computing infrastructure that encompasses the key elements of today’s networking technologies, such as systems and network management; the Internet and intranets; clients and servers; application programs; databases; transaction processing; and various operating systems and communication protocols.

**Network File System (NFS).** A protocol developed by Sun Microsystems, Incorporated, that allows any host in a network to mount another host’s file directories. Once mounted, the file directory appears to reside on the local host.

**network gateway accounting (NGA).** The NetView Performance Monitor (NPM) subsystem that receives traffic information from the gateway NCP for sessions that flow throughout a network.

**Network Information Center (NIC).** In Internet communications, local, regional, and national groups throughout the world who provide assistance, documentation, training, and other services to users.

**Network Information Services (NIS).** A set of UNIX network services (for example, a distributed service for retrieving information about the users, groups, network addresses, and gateways in a network) that resolve naming and addressing differences among computers in a network.

**network log.** A file that contains (a) messages, commands, and command procedures that have been processed by Tivoli NetView for OS/390 and (b) output resulting from commands, command procedures, and other activity occurring within Tivoli NetView for OS/390.

**network management gateway (NMG).** A gateway between Tivoli NetView for OS/390, which is the SNA network management system, and the network management function of one or more non-SNA networks.

**network management vector transport (NMVT).** A management services request/response unit (RU) that flows over an active session between physical unit management services and control point management services (SSCP-PU session).

**Network News Transfer Protocol (NNTP).** In the Internet suite of protocols, a protocol for the distribution, inquiry, retrieval, and posting of news articles that are stored in a central database.

**network session accounting (NSA).** The NetView Performance Monitor (NPM) subsystem that receives session accounting information from the NCP for sessions that flow throughout a network.

**network topology database.** The representation of the current connectivity between the network nodes within an APPN network. It includes (a) entries for all network nodes and the transmission groups interconnecting them and (b) entries for all virtual routing nodes to which network nodes are attached.

**NFS.** See Network File System

**NFS client.** A program or system that mounts remote file directories from another host called a Network File System (NFS) server.

**NFS server.** A program or system that allows authorized remote hosts called Network File System (NFS) clients to mount and access its local file directories.

**NGA.** See network gateway accounting

**NGMF.** See NetView Graphic Monitor Facility

**NIAF/2.** See NetView Installation and Administration Facility/2

**NIC.** See Network Information Center

**NIS.** See Network Information Services

**NLDM.** In Tivoli NetView for OS/390, a command that starts the session monitor. Also, the use of the abbreviation “NLDM” indicates that various panels and functions are part of the session monitor.

**NMG.** See network management gateway

**NMVT.** See network management vector transport

**NNT.** See NetView-NetView task

**NNTP.** See Network News Transfer Protocol

**non-generic alert.** In SNA management services (SNA/MS), alert information that is encoded such that it conveys to the receiver the set of screens that should be displayed for the network operator when the alert is received. The use of non-generic alerts requires that the receiver recognize and understand each unique problem for which an alert is sent. Contrast with generic alert.

**NOS.** Network operating system.

**notice.** In a Tivoli environment, a message generated by a systems management operation that contains information about an event or the status of an application. Notices are stored in notice groups. See bulletin board.
notice group. In a Tivoli environment, an application- or operation-specific container that stores and displays notices pertaining to specific Tivoli functions. The Tivoli bulletin board is comprised of notice groups. A Tivoli administrator can subscribe to one or more notice groups; the administrator’s bulletin board contains only the notices that reside in a notice group to which the administrator is subscribed.

notification. (1) An unscheduled, spontaneously generated report of an event that has occurred. (2) In systems management, information emitted by a managed object relating to an event that has occurred within the managed object, such as a threshold violation or a change in configuration status.

NPALU. In the NetView Performance Monitor (NPM), the virtual logical unit generated in an NCP with which the network subsystem communicates.

NPDA. In Tivoli NetView for OS/390, a command that starts the hardware monitor. Also, the use of the abbreviation “NPDA” indicates that various panels and functions are part of the hardware monitor.

NPM. See NetView Performance Monitor.

NSA. See network session accounting.

NT repeater. In a Tivoli environment, the first Windows NT machine on which the Tivoli Remote Execution Service is installed. Using fanout, the NT repeater distributes the Tivoli Remote Execution Service to all other NT clients during the client installation process.

null resource. In the NetView Graphic Monitor Facility, an object that is used only as an aid in formatting and drawing a view. A null resource always shows the status “unknown.”

O

object. (1) In object-oriented design or programming, a concrete realization of a class that consists of data and the operations associated with that data. (2) An item that a user can manipulate as a single unit to perform a task. An object can appear as text, an icon, or both. (3) In Tivoli NetView for OS/390, the part of a NetView command that follows the verb. The object describes where the action of the verb is to occur.

object dispatcher. See object request broker.

object identifier (OID). An administratively assigned data value of the type defined in abstract syntax notation 1 (ASN.1).

Object Management Group (OMG). A non-profit consortium whose purpose is to promote object-oriented technology and the standardization of that technology. The Object Management Group was formed to help reduce the complexity, lower the costs, and hasten the introduction of new software applications.

object path. In a Tivoli environment, an absolute or relative path to a Tivoli object, similar to paths in file systems.

object reference. In a Tivoli environment, the object identifier (OID) given to an object during its creation.

object registration service (ORS). In Tivoli NetView, a component that creates and maintains a global directory of object managers, their locations, and their protocols. The postmaster daemon uses this directory to route messages and provide location transparency for managers and agents.

object request broker (ORB). In object-oriented programming, software that serves as an intermediary by transparently enabling objects to exchange requests and responses. See Common Object Request Broker Architecture.

ODBC. See Open Database Connectivity.

OID. See object identifier.

OMG. See Object Management Group.

Open Database Connectivity (ODBC). A standard application programming interface (API) for accessing data in both relational and nonrelational database management systems. Using this API, database applications can access data stored in database management systems on a variety of computers even if each database management system uses a different data storage format and programming interface. ODBC is based on the call level interface (CLI) specification of the X/Open SQL Access Group and was developed by Digital Equipment Corporation (DEC), Lotus®, Microsoft, and Sybase. Contrast with Java Database Connectivity.

operation. In object-oriented design or programming, a service that can be requested at the boundary of an object. Operations include modifying an object or disclosing information about an object.

operations and administration. The Tivoli management discipline that addresses the automation of activities that ensure the operational integrity and reliability of a network computing system. See availability management, deployment management, and security management.

operator. A person or a program that manages activities that are controlled by a specific computer program.

operator profile. In Tivoli NetView for OS/390, a specification of the resources and activities over which a network operator has control. The profile is stored in a file that is activated when the operator logs on.
operator station task (OST). In Tivoli NetView for OS/390, the task that establishes and maintains the online session with the network operator. There is one operator station task for each network operator who logs on to Tivoli NetView for OS/390. See NetView-NetView task.

ORB. See object request broker.

ORS. See object registration service.

oserv. The name of the object request broker used by the Tivoli environment. Oserv runs on the TMR server and each TMR client.

OST. See operator station task.

package definition file (PDF). In Tivoli IT Director, an ASCII text file that contains predefined workstation, sharing, and inventory property settings for a file package.

packet. In data communication, a sequence of binary digits, including data and control signals, that is transmitted and switched as a composite whole. The data, control signals, and, possibly, error control information are arranged in a specific format. (1)

parameter. (1) A variable that is given a constant value for a specified application and that may denote the application. (1) (A) (2) In Common User Access (CUA®) architecture, a variable used in conjunction with a command to affect its result. (3) An item in a menu for which the user specifies a value or for which the system provides a value when the menu is interpreted. (4) Data passed to a program or procedure by a user or another program, namely as an operand in a language statement, as an item in a menu, or as a shared data structure. (5) In Tivoli NetView for OS/390, a part of a command's object. (6) See keyword and keyword parameter.

parent process. In the UNIX operating system, a process that creates other processes. See child process.

parent resource. In the NetView Graphic Monitor Facility, a resource that has one or more child resources below it in a hierarchy.

PassTicket. In RACF® secured sign-on, a dynamically generated, random, one-time-use, password substitute that a workstation or other client can use to sign on to the host rather than sending a RACF password across the network.

PassTicket application key. In RACF secured sign-on, an encryption key that is used in the creation and evaluation of a PassTicket. The PassTicket application key is sometimes referred to as the "secured sign-on application key."

path. (1) A list of one or more directory names and an object name (such as the name of a file) that are separated by an operating system-specific character, such as the slash (/) in UNIX operating systems, the backslash (\) in Windows operating systems, and the semicolon (;) in OS/2 operating systems. The directory names detail the path to follow, in left-to-right order, to locate the object within the file system. This concept of path is also known as the "pathname." (2) A list of directory names, usually separated by a colon (:), that are to be searched (in left-to-right order) to locate an object. This concept of path is also known as the "search path." (3) See absolute path, directory, relative path, root directory, and working directory.

pathname. See path.

path test. A test provided by Tivoli NetView for OS/390 that enables a network operator to determine whether a path is available between two LUs that are currently in session.

pattern-matching character. A special character such as an asterisk (*) or a question mark (?) that can be used to represent one or more characters. Any character or set of characters can replace a pattern-matching character.

PC agent. In a Tivoli environment, software installed on a client PC that enables Tivoli operations to execute on the PC. See PC managed node.

PC managed node. In a Tivoli environment, an object that represents a client PC. The Tivoli Management Framework can communicate with the client PC only if the PC agent is installed on the PC. Client PCs are most often referred to as PC managed nodes.

PDF. (1) See package definition file (2) See Portable Document Format.

performance class. In Tivoli NetView for OS/390, a description of an objective or commitment of performance. It consists of a performance class name, boundary definitions, response time definition, response time ranges, and response time percentage objectives. Sessions may be assigned performance classes.

persistent LU-LU session. See persistent session.

persistent session. (1) In Tivoli NetView for OS/390, a network management session that remains active even though there is no activity on the session for a specified period of time. (2) An LU-LU session that...
VTAM retains after the failure of a VTAM application program. Following the application program’s recovery, the application program restores or terminates the session. This session is sometimes referred to as a “single-node persistent session.” See multinode persistent session.

pipeline. (1) A serial arrangement of processors or a serial arrangement of registers within a processor. Each processor or register performs part of a task and passes results to the next processor; several parts of different tasks can be performed at the same time. (2) To perform processes in series. (3) To start execution of an instruction sequence before the previous instruction sequence is completed to increase processing speed. (4) In Tivoli NetView for OS/390, a message processing procedure that consists of one or more programs known as stages.

pixel map. (1) A three-dimensional array of bits. A pixel map can be thought of as a two-dimensional array of pixels, with each pixel being a value from zero to 2 to the power N -1, where N is the depth of the pixel map. (2) In the X Window System, a data type to which icons, originally created as bitmaps, are converted.

pixmap. See pixel map.

platform. An ambiguous term that may refer to the hardware, the operating system, or a combination of the hardware and the operating system on which software programs run.

plex. A Printing Systems Manager (PSM) attribute used for defining the capability of a printer to support different placements of output images on a medium. For example, the plex attribute could specify whether the printer is to support simplex or tumble mode.

Plus module. See Tivoli Plus module.

policy. In a Tivoli environment, a set of rules that are applied to managed resources. A specific rule in a policy is referred to as a “policy method.”

policy region. In a Tivoli environment, a group of managed resources that share one or more common policies. Tivoli administrators use policy regions to model the management and organizational structure of a network computing environment. The administrators can group similar resources, define access to and control the resources, and associate rules for governing the resources. The policy region contains resource types and the list of resources to be managed. A policy region is represented on the Tivoli desktop by an icon that resembles a capitol building (dome icon). When a Tivoli Management Region (TMR) is created, a policy region with the same name is also created. In this case, the TMR has only one policy region. However, in most cases, a Tivoli administrator creates other policy regions and subregions to represent the organization of the

TMR. A TMR addresses the physical connectivity of resources whereas a policy region addresses the logical organization of resources.

policy subregion. In a Tivoli environment, a policy region created or residing in another policy region. When a policy subregion is created, it initially uses the resource and policy properties of the parent policy region. The Tivoli administrator can later change or customize these properties to reflect the specific needs and differences of the subregion.

polling. (1) On a multipoint connection or a point-to-point connection, the process whereby data stations are invited, one at a time, to transmit. (I) (2) Interrogation of devices for such purposes as to avoid contention, to determine operational status, or to determine readiness to send or receive data. (A) (3) In network management, the process by which a manager interrogates one or more managed nodes at regular intervals.

populate. In a Tivoli environment, to fill a profile with information that is to be distributed to the subscribing managed resources.

port. To modify a computer program to enable it to run on a different platform.

Portable Document Format (PDF). A standard specified by Adobe Systems, Incorporated, for the electronic distribution of documents. PDF files are compact; can be distributed globally via e-mail, the Web, intranets, or CD-ROM; and can be viewed with the Acrobat Reader, which is software from Adobe Systems that can be downloaded at no cost from the Adobe Systems home page.

portmapper. A program that maps client programs to the port numbers of server programs. Portmapper is used with remote procedure call (RPC) programs.

positional operand. (1) An operand in a language statement that has a fixed position. (2) Contrast with keyword operand. (3) See definition statement.

postmaster. In Tivoli NetView, a process (daemon) that directs network management information between multiple application programs and agents running concurrently. The postmaster determines the route by using specified addresses or a routing table that is configured in the object registration service.

PPI. See program-to-program interface.

presentation graphics feature (PGF). In the NetView Performance Monitor (NPM), a feature used in conjunction with the Graphical Data Display Manager (GDDM) to generate online graphs in the NPM graphic subsystem.
presentation services command processor (PSCP).  In Tivoli NetView, a facility that processes requests from a user terminal and formats displays to be presented at the user terminal.

primary database.  In Tivoli NetView for OS/390, the main database provided to the NetView user for recording error data. See secondary database.

primary POI task (PPT).  In Tivoli NetView for OS/390, the subtask that processes all unsolicited messages that are received from the VTAM program operator interface (POI) and delivers them to the controlling operator or to the command processor. The PPT also processes (a) the initial command that is specified to execute when NetView is initialized and (b) timer request commands that are scheduled to execute under the PPT.

primary window.  In OSF/Motif, the top-level window in an application program that can be minimized or represented by an icon. See submap window.

principal name.  (1) In Kerberos, the name by which the Kerberos principal is identified. The principal name consists of three parts: a service or user name, an instance name, and a realm name. (2) In a Tivoli environment, an operating system user ID that is associated with a Tivoli administrator.

principal password.  In Kerberos, the password that corresponds to the principal name. This password is used to authenticate services and users to each other.

print file document.  A Printing Systems Manager (PSM) object that represents text or data to be printed by a job. Contrast with print resource document.

Printing Systems Manager (PSM).  An IBM licensed program that applies print administration and management technology to a cross-platform, client/server print system. PSM provides a set of (a) printing functions for submitting and controlling print jobs and (b) systems management and operator functions to control print spoolers and print supervisors. PSM is based on the Palladium distributed print system.

print resource document.  A Printing Systems Manager (PSM) object that represents a resource, such as graphics or fonts, used by a job to print a print file document.

Print Services Facility™ (PSF) for AIX.  An IBM licensed printer driver program that produces printer commands from the data sent to it.

private key.  In computer security, a key that is known only to its owner. Contrast with public key. See public key cryptography.

profile.  In a Tivoli environment, a container for application-specific information about a particular type of resource. A Tivoli application specifies the template for its profiles; the template includes information about the resources that can be managed by that Tivoli application.

A profile is created in the context of a profile manager; the profile manager links a profile to the Tivoli resource (for example, a managed node) that uses the information contained in the profile. A profile does not have any direct subscribers.

profile manager.  In a Tivoli environment, a container for profiles that links the profiles to a set of resources, called "subscribers." A profile manager can contain (a) profiles of multiple types or (b) multiple profiles of the same type. Tivoli administrators use profile managers to organize and distribute profiles. A profile manager is created in the context of a policy region and is a managed resource in a policy region. See subscription list.

program-to-program interface (PPI).  In Tivoli NetView for OS/390, a facility that allows user programs to send data buffers to or receive data buffers from other user programs. It also allows system and application programs to send alerts to the NetView hardware monitor.

prototype profile.  In a Tivoli environment, a model profile from which a Tivoli administrator can create other profiles, often by cloning the prototype profile.

proxy endpoint.  In Tivoli Distributed Monitoring, a representation for an entity (such as a network device or a host) that functions as a subscriber for Tivoli Distributed Monitoring profiles. A Tivoli administrator associates each proxy endpoint with a managed node; several proxy endpoints can be associated with a single managed node.

PSCP.  See presentation services command processor.

PSF.  Print Services Facility. See Print Services Facility for AIX.

PSM.  See Printing Systems Manager.

public key.  In computer security, a key that is made available to everyone. Contrast with private key. See public key cryptography.

public key cryptography.  In computer security, cryptography in which public keys and private keys are used for encryption and decryption.

pull.  A network operation that initiates an action by requesting the action from a resource. Contrast with push.

push.  A network operation that sends information to resources. Contrast with pull.
query. In a Tivoli environment, a combination of statements that are used to search the configuration repository for systems that meet certain criteria.

query library. In a Tivoli environment, a facility that provides a way to create and manage Tivoli queries.

R

RACF. See Resource Access Control Facility.

RACF secured sign-on. In the Resource Access Control Facility (RACF), a function that enables workstations and other clients to sign on to the host and communicate in a secure way without having to send RACF passwords across the network. See PassTicket and PassTicket application key.

RDBMS. See relational database management system.

RDBMS Interface Module (RIM). In the Tivoli Management Framework, the module in the distributed object database that contains information about the installation of the relational database management system (RDBMS).

real object. In the NetView Graphic Monitor Facility, an object that represents an actual resource. See aggregate object.

real resource. (1) In VTAM, a resource identified by its real name and its real network identifier. (2) In the NetView Graphic Monitor Facility, an individual network resource represented by a real object.

recommended action. The procedures that Tivoli NetView for OS/390 recommends for determining and correcting the causes of network problems.

recording filter. In Tivoli NetView for OS/390, the function that determines which events, statistics, and alerts are stored in a database.

reference implementation. An implementation by which other implementations are judged for conformance to a standard or are tested for interoperability.

reference model. In the context of Tivoli software, the model configuration for a system or set of systems that is used to maintain consistent configurations in a distributed environment. In Tivoli Inventory, reference models are created in the configuration repository.

registered name. In a Tivoli environment, the name by which a particular resource is registered with the name registry when it is created.

registration file. See application registration file, field registration file, local registration file, and symbol registration file.

regular command. See immediate command.

relation. (1) In a relational database, a set of entity occurrences that have the same attributes. (T) (2) The comparison of two expressions to see if the value of one is equal to, less than, or greater than the value of the other. (3) In a relational database, a table that identifies entities and their attributes.

relational database. A database in which the data are organized and accessed according to relations. (T)

relational database management system (RDBMS). A collection of hardware and software that organizes and provides access to a relational database.

relative path. A path that begins with the working directory. Contrast with absolute path.

remote distribution. In a Tivoli environment, a distribution to target machines in a connected Tivoli Management Region.

remote host. Any host on a network except the host at which a particular operator is working.

Remote Operations Service (ROPS). In Communications Server, an application program on a client workstation that processes commands that are issued by Tivoli NetView for OS/390 through the Service Point Application (SPA) Router, thus enabling Tivoli NetView for OS/390 to manage distributed networks and application programs.

remote procedure call (RPC). (1) A facility that a client uses to request the execution of a procedure call from a server. This facility includes a library of procedures and an external data representation. (2) A client request to a service provider located in another node.

repeater. (1) A node of a local area network; a device that regenerates signals in order to extend the range of transmission between data stations or to interconnect two branches. (T) (2) See repeater site.

repeater range. In a Tivoli environment, the Tivoli clients that receive data from the repeater site.

repeater site. In a Tivoli Management Region, a managed node that is configured with the MDist feature. A repeater site receives a single copy of data and distributes it to the next tier of clients.

requester. See client.

Request for Comments (RFC). In Internet communications, the document series that describes a
part of the Internet suite of protocols and related experiments. All Internet standards are documented as RFCs.

**resource.** (1) Any facility of a computing system or operating system required by a job or task, and including main storage, input/output devices, the processing unit, data sets, and control or processing programs. (2) In Tivoli NetView for OS/390, any hardware or software that provides function to the network. (3) See managed resource.

**Resource Access Control Facility (RACF).** An IBM licensed program that provides for access control by identifying and verifying the users of the system, by authorizing access to protected resources, by logging the detected unauthorized attempts to enter the system, and by logging the detected accesses to protected resources.

**resource label.** In the NetView Graphic Monitor Facility, the textual information that identifies a particular aggregate or real resource. The resource label is displayed next to the resource symbol and cannot be changed by the network operator.

**resource level.** In Tivoli NetView for OS/390, the hierarchical position of a device (and the software contained within it) in a data processing system. For example, a first-level resource could be the communication controller, and the second-level resource could be the line connected to it.

**resource manager.** In Tivoli NetView for OS/390, an application program that manages specific network resources. Each resource manager is assigned a unique range of command indicators that specify the command support characteristics for the resources that it manages. The resource manager provides information to the NetView Graphic Monitor Facility (NGMF).

**Resource Object Data Manager (RODM).** In Tivoli NetView for OS/390, a component that operates as a cache manager and that supports automation applications. RODM provides an in-memory cache for maintaining real-time data in an address space that is accessible by multiple applications.

**resource resolution table (RRT).** In NetView Performance Monitor (NPM), a table that contains the names of the network resources for which data is to be collected. The NPM RRT corresponds with an NCP and is built by NPMGEN from an NCP Stage I and an NCP RRT.

**resource status collector.** In Tivoli NetView for OS/390, a function that collects status information on monitored resources and forwards this information to the resource status manager.

**resource status manager.** The part of the NetView Graphic Monitor Facility that maintains a database of SNA resource status information and that forwards this information to all attached server workstations.

**resource symbol.** In the NetView Graphic Monitor Facility, a geometric shape (such as a line, square, or octagon) that represents a particular kind of resource and indicates whether that resource is one resource or a composite of a group of resources.

**resource type.** (1) In a Tivoli environment, one of the properties of a managed resource. Resource types are defined in the default policy for a policy region. (2) In Tivoli NetView for OS/390, one of the three elements, which also include data type and display type, that are used to describe the organization of panels. Resource types in one category include central processing unit, channel, control unit, and I/O device; and in another category, they include communication controller, adapter, link, cluster controller, and terminal.

**response level.** See alarm level.

**response time.** (1) The elapsed time between the end of an inquiry or demand on a computer system and the beginning of the response; for example, the length of time between an indication of the end of an inquiry and the display of the first character of the response at a user terminal. (2) (A) (2) For response time monitoring, the time from the activation of a transaction until a response is received, according to the response time definition coded in the performance class.

**response time monitor (RTM).** A feature available with certain hardware devices to allow measurement of response times, which may be collected and displayed by Tivoli NetView for OS/390.

**review file.** In the NetView Performance Monitor (NPM), a VSAM key-sequenced data set (KSDS) containing data collected and recorded as a result of a network start display command or start monitor command.

**RFC.** See Request for Comments.

**RIM.** See RDBMS Interface Module.

**RIM repository.** See configuration repository.

**RODM.** See Resource Object Data Manager.

**RODM-based view.** In the NetView Graphic Monitor Facility (NGMF), a view that is predefined or dynamically built based on definitions in RODM. Examples of a RODM-based view are network views, exception views, configuration views, and more-detail views.

**RODM resource.** In the context of NetView Graphic Monitor Facility (NGMF) views, an object created in RODM to represent a resource. These objects can be created by loader files, user applications, or by the SNA topology manager.
role. See [authorization role].

root directory. The highest level directory in a hierarchical file system.

root user. In the UNIX operating system, a user who has superuser authority.

ROPS. See [Remote Operations Service].

router. (1) A computer that determines the path of network traffic flow. The path selection is made from several paths based on information obtained from specific protocols, algorithms that attempt to identify the shortest or best path, and other criteria such as metrics or protocol-specific destination addresses. (2) An attaching device that connects two LAN segments, which use similar or different architectures, at the reference model network layer. (3) In OSI terminology, a function that determines a path by which an entity can be reached. (4) Contrast with bridgel.

RPC. See [remote procedure call].

RRT. See [resource resolution table].

RS/6000®. A family of workstations and servers based on IBM's POWER architecture. They are primarily designed for running multiluser numerical computing applications that use the AIX operating system.

RTM. See [response time monitor].

rule. In the Tivoli Enterprise Console, a set of one or more logical statements that enable the event server to recognize relationships among events (event correlation) and to execute automated responses accordingly.

rule base. In the Tivoli Enterprise Console, a set of rules and the event class definitions for which the rules are written. The Tivoli Enterprise Console uses the rule base in managing events. An organization can create many rule bases, with each rule base fulfilling a different set of needs for network computing management.

S

SAF. See [System Authorization Facility].

scalable. Pertaining to the capability of a system to adapt readily to a greater or lesser intensity of use, volume, or demand. For example, a scalable system can efficiently adapt to work with larger or smaller networks performing tasks of varying complexity.

scanner. In a Tivoli environment, the software installed on each PC managed node that is to be scanned by Tivoli Inventory.

scheduler. A computer program designed to perform functions such as scheduling, initiation, and termination of jobs. (A)

schema. The set of statements, expressed in a data definition language, that completely describe the structure of a database.

scope check. In Tivoli NetView for OS/390, the process of verifying that an operator is authorized to issue a particular command. Contrast with span check. See [scope of command authorization].

scope of command authorization. In Tivoli NetView for OS/390, the level of access authority that a system programmer or system administrator grants to a network operator to use various commands. See [scope check].

script. (1) A computer program that is interpreted. (2) See [shell script].

script stub. A placeholder for a particular shell script. For example, the Tivoli Module Builder generates a script stub (using a skeleton file) if a developer does not provide the script for implementing a particular task or monitor when defining the task or monitor; the script stub then displays a message that the script executed successfully and displays any variables that were passed to the script.

search path. See [path].

seat. A slang term that refers to the number of licensed users of a software product, which is the same as the number of installations of the product. For example, if there were 100 Lotus Notes™ seats, there would be 100 licensed users of Lotus Notes (or 100 installations of Lotus Notes).

secondary database. One of two databases provided by Tivoli NetView for OS/390 for recording data. It provides backup or a temporary storage alternative to the primary database. See [primary database].

security group. In a Tivoli environment, a group of managed resources over which a Tivoli administrator is granted authority. Examples of a security group include a policy region and the administrator collection.

security management. The Tivoli management discipline that addresses the organization’s ability to control access to applications and data that are critical to its success. See [availability management], [deployment management], and [operations and administration].

seed file. In Tivoli NetView, a file that contains a list of nodes within an Administrative Domain, which the automatic discovery function uses to accelerate the generation of the network topology map.

segment. (1) A portion of a computer program that may be executed without the entire computer program being resident in main storage. (T) (2) A group of display elements. (3) A section of cable between components or devices. A segment may consist of a single patch cable, several patch cables that are...
connected, or a combination of building cable and patch cables that are connected. (4) In the Enhanced X-Windows Toolkit, one or more lines that are drawn but not necessarily connected at the endpoints. (5) In LANs or WANs, a subset of nodes in a network or subnet that are connected by a common physical medium.

**senior role.** See [authorization role](#).

**server.** A functional unit that provides services to one or more clients over a network. Examples include a file server, a print server, and a mail server.

**server workstation.** In the NetView Graphic Monitor Facility, a workstation with the graphic data server. This workstation uses the graphic monitor and the view administrator for administrative functions. The server workstation sends status information to client workstations over an LU 6.2 session.

**Service Level Reporter (SLR).** A licensed program that generates management reports from data sets such as System Management Facility (SMF) files.

**service point (SP).** An entry point that supports applications providing network management for resources that are not under its direct control as an entry point. Each resource is either under the direct control of another entry point or not under the direct control of any entry point. A service point accessing these resources is not required to use SNA sessions (unlike a focal point).

**Service Point Application Router.** In Communications Server, software that receives commands issued from Tivoli NetView for OS/390 and sends these commands to an application program, called the Remote Operations Service (ROPS), to be processed on a client workstation.

**service point command facility (SPCF).** A program or function that exchanges data and control between the network operator, the link connection component manager (LCCM), and the link connection subsystem manager (LCSM).

**service point command service (SPCS).** In Tivoli NetView for OS/390, an extension of the command facility that allows the host processor to communicate with a service point by using the communication network management (CNM) interface.

**session data.** Session awareness data, session trace data, and session response time data that Tivoli NetView for OS/390 collects.

**session manager (SM).** A product, such as NetView Access Services, that allows a user at a terminal to log on to multiple applications concurrently.

**session monitor.** In Tivoli NetView for OS/390, the component that collects and correlates session-related data and provides online access to this information. Contrast with [hardware monitor](#).

**session setup failure notification (SSFN).** In Tivoli NetView for OS/390, session awareness data that is provided when there is a failure. It identifies the system services control point (SSCP) that detects the error, the SSCP(s) that are involved, and the names of the session partners affected.

**session statistics file.** In the NetView Performance Monitor (NPM), an online VSAM key-sequenced data set (KSDS) used for storing session data.

**session trace.** In Tivoli NetView for OS/390, the function that collects session trace data for sessions involving specified resource types or involving a specific resource.

**session trace data.** Data, relating to sessions, that is collected by Tivoli NetView for OS/390 whenever a session trace is started and that consists of session activation parameters, VTAM path information unit (PIU) data, and NCP data.

**severity level.** In the Tivoli Enterprise Console, a classification for an event that indicates its degree of severity. Severity levels can be modified by a user or a Tivoli Enterprise Console rule. The predefined severity levels, in order of descending severity, include: fatal, critical, warning, minor, harmless, and unknown.

**shared application program.** In Tivoli NetView, an application program that serves multiple action requests; however, only one instance of the application program can run in a given graphical user interface (GUI).

**shared submap.** In Tivoli NetView, a submap on which multiple application programs manage objects on the application plane. Shared submaps allow application programs to cooperatively contribute information to the same submap. Contrast with [exclusive submap](#).

**shell.** A software interface between a user and the operating system of a computer. Shells interpret commands and user interactions on devices such as keyboards, pointing devices, and touch-sensitive screens and communicate them to the operating system. Shells simplify user interactions by eliminating the user’s concern with operating system requirements. A computer may have several layers of shells for various levels of user interaction.

**shell procedure.** See [shell script](#).

**shell prompt.** In the UNIX operating system, the character string on the command line indicating that the system can accept a command (typically the `$` character).

**shell script.** In the UNIX operating system, a series of commands, combined in a file, that carry out a particular...
function when the file is run or when the file is specified as a value to the SH command.

**show cause.** The reason code in the record maintenance statistics (RECMS) that indicates to VTAM or to Tivoli NetView for OS/390 the threshold that was exceeded and whether the threshold has been dynamically altered.

**SIA.** System Information Agent. See [Tivoli Distributed Monitoring](https://www.ibm.com) the product that replaces the System Information Agent.

**signal.** In computer software, a message that is sent to a process to change its behavior based on the value sent to it.

**signature.** In computer software, the name of an operation and its parameters.

**simple connection.** In Tivoli NetView, the representation of connectivity as seen from one endpoint of a connection.

**Simple Network Management Protocol (SNMP).** In the Internet suite of protocols, a network management protocol that is used to monitor routers and attached networks. SNMP is an application layer protocol. Information on devices managed is defined and stored in the application’s Management Information Base (MIB).

**singular filter.** A filter that identifies a host, subnet, or all hosts with a single expression.

**skeleton file.** A program template that the Tivoli Module Builder uses to generate any text-based file, including scripts, Java or C source files, build files, and help text. A skeleton file includes substitution variables that are replaced at run time. The values for these variables originate from user-defined variables or values specified in a component description file (CDF) or a global description file (GDF) file.

**SLR.** See [Service Level Reporter](https://www.ibm.com)

**SMF.** See [System Management Facility](https://www.ibm.com)

**SMIT.** See [System Management Interface Tool](https://www.ibm.com)

**SMS.** See [Storage Management Subsystem](https://www.ibm.com)

**snapshot.** In Tivoli NetView, a copy of a map that reflects the topology and status of the map’s nodes and links at a given moment in time.

**SNATM.** See [SNA topology managed](https://www.ibm.com)

**SNA topology manager (SNATM).** In Tivoli NetView for OS/390, a component that dynamically collects status and topology data into the Resource Object Data Manager (RODM) for display by the NetView Graphic Monitor Facility (NGMF). SNATM includes the function formerly provided by the APPN Topology and Accounting Manager (APPNTAM) feature of NetView for MVS V2R4.

**SNMP.** See [Simple Network Management Protocol](https://www.ibm.com)

**Software Installer for OS/2.** An OS/2-based tool that is used to install workstation functions such as the NetView Graphic Monitor Facility.

**software management.** See [change management](https://www.ibm.com)

**source host.** In Tivoli Software Distribution, the managed node on which the files and directories referenced in a file package reside.

**span.** In Tivoli NetView for OS/390, a user-defined group of network resources within a single domain. Spans provide a level of security by allowing the system administrator to define (a) the resources to which an operator can issue commands, (b) the views of resources that an operator can display, and (c) the resources in a view that an operator is allowed to see (an operator may not be authorized to see all the resources in a particular view). See [span check](https://www.ibm.com).

**span check.** In Tivoli NetView for OS/390, the process of verifying that an operator is authorized to perform actions on a network resource, a NetView Graphic Monitor Facility (NGMF) view, or a resource within a view. Contrast with [scope check](https://www.ibm.com).

**SPA Router.** See [Service Point Application Router](https://www.ibm.com)

**SPCF.** See [service point command facility](https://www.ibm.com)

**SQL.** A programming language that is used to define and manipulate data in a relational database.

**SRF.** See [symbol registration file](https://www.ibm.com)

**SSFN.** See [session setup failure notification](https://www.ibm.com)

**SSI.** See [subsystem interface](https://www.ibm.com)

**stage.** In Tivoli NetView for OS/390, a program that processes messages in a NetView pipeline. Stages send messages to each other serially.

**statistics record.** In Tivoli NetView for OS/390, a resource-generated database record that contains various statistics about a resource.

**status monitor.** In Tivoli NetView for OS/390, a component that collects and summarizes information on the status of resources defined in a VTAM domain.

**Storage Management Subsystem (SMS).** A DFSMS/MVS facility that is used to automate and centralize the management of storage. Using SMS, a storage administrator describes data allocation characteristics, performance and availability goals, backup and retention requirements, and storage requirements to the system.
structured query language. see sql.

subagent. in the simple network management protocol (snmp), something that provides an extension to the utility provided by the snmp agent.

submap. in tivoli netview, a particular view of some aspect of a network that displays symbols representing objects. the application program that creates a submap determines what part of the network the submap displays.

submap pane. the area of a submap window in which the submap is displayed.

submap stack. in tivoli netview, a component of the graphical user interface shown on the left side of each submap window. the submap stack represents the navigational path used to reach the particular submap, and it can be used to select a previously viewed submap.

submap window. in tivoli netview, the graphical component that contains a menu bar, a submap viewing area, a status line, and a button box. a user can display multiple submap windows of an open map and an open snapshot at any given time. see primary window.

subnet. any group of nodes that have a set of common characteristics, such as the same network id.

subscriber. in a tivoli environment, a tivoli client, a profile manager, or any endpoint type (for example, a pc managed node or a proxy endpoint) that is subscribed to a profile manager. although profiles are distributed to a subscriber, the subscriber may or may not be the final destination of the profile distribution.

subscription. in a tivoli environment, the process of identifying the subscribers to which profiles will be distributed.

subscription list. in a tivoli environment, a list that identifies the subscribers to a profile manager. including a profile manager on a subscription list (in effect, a list within a list) is a way of subscribing several resources simultaneously rather than adding each one individually.

in tivoli plus modules, a profile manager functions as a subscription list.

subsystem interface (ssi). the mvs interface by which routines (ibm-, vendor-, or installation-written) request services of, or pass information to, subsystems. the ssi is used by tivoli netview for os/390 to receive system messages and enter system commands (when used with extended mcs consoles, it is used to receive commands, not messages), and to communicate with other instances of tivoli netview for os/390.

super role. see authorization role.

superuser authority. in the unix operating system, the unrestricted authority to access and modify any part of the operating system, usually associated with the user who manages the system. see root user.

suppression character. in tivoli netview for os/390, a user-defined character that is coded at the beginning of a command list statement or a command to prevent the statement or command from appearing on the operator's terminal screen or in the network log.

symbol. in tivoli netview, a picture or an icon on a submap that represents an object (a network resource or an application). each symbol belongs to a class, represented by the symbol's shape, and to a subclass, represented by the design within the shape. the symbol reflects characteristics of the object it represents, such as its status; it also has characteristics of its own, such as behavior.

symbol registration file (srf). in tivoli netview, a file used to define symbol classes and subclasses.

synchronous monitor. in tivoli distributed monitoring, a monitor that monitors resources on a periodic basis (most monitors are synchronous). contrast with asynchronous monitor.

sysplex. a set of mvs or os/390 systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads. this term is derived from "system complex."

system authorization facility (saf). an interface defined by mvs that enables programs to use system authorization services in order to protect access to resources such as data sets and mvs commands. the ibm resource access control facility (racf) is a product that uses the saf interface.

system configuration. a process that specifies the devices and programs that form a particular data processing system.

system information agent (sia). see tivoli distributed monitoring, the product that replaces the system information agent.

system management facility (smf). a standard feature of os/390 that collects and records a variety of system and job-related information.

system management interface tool (smitt). an interface tool of the aix operating system for installing, maintaining, configuring, and diagnosing tasks.

systems management. (1) functions in the application layer related to the management of open systems interconnection (osi) resources and their status across all layers of the osi architecture. (2) the tasks involved in maintaining computer and communication systems, for example: changing
configuration, identifying faults, securing access, accounting for resource usage, and analyzing performance.

T

TACF. See Tivoli Access Control Facility.

TACF database. In Tivoli Security Management, a database that contains the customized rules that the authorization daemon for the Tivoli Access Control Facility (TACF) uses to allow or to deny resource accesses in the UNIX environment.

TACF lookaside database. In Tivoli Security Management, a database that provides ID-to-name resolution, thereby enabling the Tivoli Access Control Facility (TACF) to convert UNIX IDs (user IDs, group IDs, IP addresses, and port numbers) to names at run time.

TAF. See terminal access facility.

TAP. See Telocator Alphanumeric Protocol.

target. See endpoint and execution target.

target host. See endpoint.

task. (1) In a multiprogramming or multiprocessing environment, one or more sequences of instructions treated by a control program as an element of work to be accomplished by a computer. (2) In a Tivoli environment, the definition of an action that must be routinely performed on various managed nodes throughout the network. A task defines the executables to be run when the task is executed, the authorization role required to execute the task, and the user or group name under which the task will execute.

task endpoint. See endpoint.

task library. In a Tivoli environment, a container in which a Tivoli administrator can create and store tasks and jobs.

Task Library Language (TLL). In a Tivoli environment, a programming language used to define a task library. The TLL definition can be used to copy a task library from one installation to another. The TLL also allows the arguments for each task to be described such that graphical user interface (GUI) tools can interpret them and present an interface for operators who want to create the tasks.


Telocator Alphanumeric Protocol (TAP). An industry standard protocol for the input of paging requests.

terminal access facility (TAF). In Tivoli NetView for OS/390, a facility that allows a network operator to control a number of subsystems. In a full-screen or operator control session, operators can control any combination of such subsystems simultaneously.

threshold. In software products, a value that defines a limit for a monitored condition. The monitored condition, the significance of the limit, and the particular software product's response when the monitored condition reaches the specified threshold vary widely according to product.

throttle. In Tivoli NetView, a condition defined in the filter table and used to regulate the flow of traps.

time to live (TTL). A technique used by best-effort delivery protocols to inhibit endlessly looping packets. The packet is discarded if the TTL counter reaches 0.

Tivoli Access Control Facility (TACF). In Tivoli Security Management, an object-oriented security system that runs on UNIX-based operating systems and provides security functions that are not available on UNIX (such as an access rule database, an audit log, and administration tools). TACF is invoked immediately after the operating system has completed its initialization, and it places hooks in system services that should be protected, thereby enabling control to be passed to TACF before the services are performed.

Tivoli administrator. In a Tivoli environment, a system administrator who has been authorized to perform systems management tasks and manage policy regions in one or more networks. Each Tivoli administrator is represented by an icon on the Tivoli desktop.

Tivoli Application Development Environment. A Tivoli toolkit that contains the complete application programming interface (API) for the Tivoli Management Framework. This toolkit enables customers and Tivoli Partners to develop their own applications for the Tivoli environment.

Tivoli Application Extension Facility. A Tivoli toolkit that enables customers to extend the capabilities of Tivoli applications. For example, customers can add fields to a dialog, create custom attributes and methods for application resources, or create custom icons and bitmaps.

Tivoli client. A client of a Tivoli server. See TMR client and TMR server.

Tivoli Cross-Site®. The integrated suite of Tivoli products for managing an e-commerce environment to ensure that Web resources are secure and available and to enable applications and information to be distributed and maintained across the extended enterprise.

Tivoli Decision Support. A Tivoli product that consolidates, transforms, and presents IT data in many different views, enabling an enterprise to gain insight.
into patterns and relationships among the data and to make critical business decisions based on this data.

**Tivoli desktop.** In a Tivoli environment, the desktop that system administrators use to manage their network computing environment.

**Tivoli Developer Kit.** See [Tivoli Module Designer](#).

**Tivoli Distributed Monitoring.** A Tivoli product that monitors system resources, initiates any necessary corrective actions, and informs system administrators of potential problems. Tivoli Distributed Monitoring consists of a group of monitors that are installed on each managed node that is to be monitored. It resolves some events on its own and may send others to the Tivoli Enterprise Console.

**Tivoli Enterprise Console.** A Tivoli product that collects, processes, and automatically initiates corrective actions for system, application, network, and database events; it is the central control point for events from all sources. The Tivoli Enterprise Console provides a centralized, global view of the network computing environment; it uses distributed event monitors to collect information, a central event server to process information, and distributed event consoles to present information to system administrators.

**Tivoli Enterprise software.** The integrated suite of Tivoli products for systems management in a large organization. These products enable system administrators to manage their network computing enterprise according to the disciplines of availability management, deployment management, operations and administration, security management, and service-level management. This suite includes Tivoli Global Enterprise Manager, Tivoli NetView for OS/390, and Tivoli Decision Support.

**Tivoli environment.** The Tivoli applications, based upon the Tivoli Management Framework, that are installed at a specific customer location and that address network computing management issues across many platforms. In a Tivoli environment, a system administrator can distribute software, manage user configurations, change access privileges, automate operations, monitor resources, and schedule jobs.

**Tivoli Event Integration Facility.** A Tivoli toolkit that provides a simple application programming interface (API) to enable customers and Tivoli Partners to develop new event adapters that can forward events to the Tivoli Enterprise Console. A customer can also translate events from third-party or in-house applications.

**Tivoli GEM.** See [Tivoli Global Enterprise Managed](#).

**Tivoli GEM module.** In a Tivoli environment, a management module that enables a particular application or business system to be managed by the Tivoli Global Enterprise Manager (Tivoli GEM).

**Tivoli Global Enterprise Manager (Tivoli GEM).** A Tivoli product that allows system administrators to graphically monitor, control, and configure applications residing in distributed and host (S/390®) environments and to use the concept of business systems management to organize related components, thereby providing a business perspective for management decisions. Tivoli Global Enterprise Manager gives information technology staff a logical view of the computing environment; this view shows, at a glance, the status of the multiple applications that comprise the enterprise’s business system, including application components, the relationships among and between components, and the flow of data between the applications. By providing this view from a business perspective, Tivoli Global Enterprise Manager enables system administrators to quickly make determinations about the business impact of any component failure.

Addressing technology problems from the business perspective greatly improves the effectiveness of system administrators and provides a higher level of service to users.

**Tivoli install image.** In a Tivoli environment, a file that resides on a CD or in a file system and contains a Tivoli product to be installed. A Tivoli install image can be used to install the Tivoli Management Framework or to install an application onto the Framework for the first time. A single CD often includes both a Tivoli install image and a Tivoli upgrade image, and it may include Tivoli install images for more than one application. Contrast with [Tivoli upgrade image](#).

**Tivoli Inventory.** A Tivoli product that enables system administrators to gather hardware and software information for a network computing environment. It scans the managed resources and stores inventory information in the configuration repository.

**Tivoli IT Director.** A Tivoli product for systems management in a small or medium organization. It is not sold directly by Tivoli Systems Inc. but rather through a Tivoli authorized reseller.

**Tivoli LAN Access.** A Tivoli product that enables system administrators to extend existing LAN management tools by integrating them with the Tivoli suite of products.

**Tivoli management agent.** In the Tivoli environment, an agent that securely performs administrative operations.

**Tivoli Management Framework.** The base software that is required to run the applications in the Tivoli product suite. This software infrastructure enables the integration of systems management applications from Tivoli Systems Inc. and the Tivoli Partners. The Tivoli Management Framework includes the following:

- Object request broker (oserv)
- Distributed object database
Basic administration functions
Basic application services
Basic desktop services such as the graphical user interface

In a Tivoli environment, the Tivoli Management Framework is installed on every client and server; however, the TMR server is the only server that holds the full object database.

Tivoli management gateway. In the Tivoli environment, a system that enables bidirectional communication with Tivoli Management Agents.

Tivoli Management Region (TMR). In a Tivoli environment, a Tivoli server and the set of clients that it serves. An organization can have more than one TMR. A TMR addresses the physical connectivity of resources whereas a policy region addresses the logical organization of resources.

Tivoli management software. The overall descriptor for software from Tivoli Systems Inc., which includes Tivoli Enterprise software (for systems management in a large organization), Tivoli IT Director (for systems management in a small or medium organization), and Tivoli Cross-Site (for the management of e-commerce systems). Tivoli management software enables organizations to centrally manage their computing resources (including the critical applications that drive business performance and profits) in a simple and straightforward manner.

Tivoli Manager. Tivoli management software that manages specific vendor systems, networks, applications, or databases.

Tivoli Module Builder (TMB). A Tivoli product that enables developers to create a special type of file, called a management module, for managing an application or business system with Tivoli management software. Management modules include Tivoli GEM modules and Tivoli Plus modules. The Tivoli Module Builder provides tools (such as the Tivoli Module Designer) and templates for describing the management characteristics of an application or business system and for building this information (together with the scripts, programs, and files that are required to implement the management function) into a Tivoli install image or an application management package. The Tivoli Module Builder uses file types defined in the Application Management Specification (AMS).

Tivoli Module Designer (TMD). A Tivoli tool that enables developers to describe the management characteristics of an application or business system and that generates the application description files and application management packages that the Tivoli management software uses to manage applications and business systems. The Tivoli Module Designer replaces the Tivoli Developer Kit.

Tivoli NetView. A Tivoli product that enables distributed network management across multiple operating systems and protocols. Unlike Tivoli NetView for OS/390, Tivoli NetView does not provide centralized management from an OS/390 host.

Tivoli NetView for OS/390. A Tivoli product that enables centralized systems and network management from an OS/390 environment. Through its MultiSystem Manager component, Tivoli NetView for OS/390 enables management of distributed resources, such as Internet Protocol (IP) resources, NetWare resources, asynchronous transfer mode (ATM) resources, and others. Contrast with Tivoli NetView.

Tivoli NetWare repeater (TNWR). In a Tivoli environment, a server application that is installed on a Novell NetWare server and that maintains a list of available clients for the server. The Tivoli NetWare repeater works with the NetWare managed site to perform profile distribution.

Tivoli Partner Association. A partnership program that is led by Tivoli Systems Inc. for business, industry, and product partners. The Tivoli Partner Association provides programs and benefits for business partners (including systems integrators, outsourcers, and resellers) to sell Tivoli Enterprise and IT Director products. Industry and product partners collaborate with Tivoli Systems Inc. in creating hardware and software products that are Tivoli Ready.

Tivoli Plus module. In a Tivoli environment, a management module that has been certified by the Tivoli Partner Association and that enables a specific vendor application to be managed by Tivoli management software. To be certified by the Tivoli Partner Association, the Tivoli Plus module must include certain features such as enablement for the Tivoli Global Enterprise Manager (Tivoli GEM).

Tivoli Ready. Pertaining to a product that has passed rigorous product certification testing by Tivoli Systems Inc. to ensure that the product delivers turnkey (or "out-of-the-box") integration with Tivoli management software. A product that has passed this certification testing carries the Tivoli Ready logo.

Tivoli Remote Control. A Tivoli product that enables a Tivoli administrator to control mouse and keyboard operations on an NT managed node or a PC managed node.

Tivoli Remote Execution Service. A service that enables a Tivoli environment to perform remote operations on machines. These operations include: remotely installing clients, connecting Tivoli Management Regions (TMRs), and starting oserv from a remote machine.

Tivoli Security Management. Tivoli Enterprise software that enables the consistent definition,
implementation, and enforcement of security policy in a network computing environment.

**Tivoli server.** The server that holds or references the complete set of Tivoli software, including the full object database. See [Tivoli client](#), [TMR client](#), and [TMR server](#).

**Tivoli Service Desk for OS/390.** A Tivoli product that is an integrated set of tools, services, and interfaces for automating and customizing an organization's IT service and support operation in an OS/390 environment. It provides a structure that supports the gathering, organizing, locating, and reporting of information related to problem, change, and asset management.

**Tivoli Software Distribution.** A Tivoli product that automates software distribution to clients and servers in a network computing environment. An organization can use this product to install and update applications and software in a coordinated, consistent manner across a network. Tivoli Software Distribution creates file packages and distributes them to predefined subscribers.

**Tivoli upgrade image.** In a Tivoli environment, a file that resides on a CD or in a file system and contains updates for a Tivoli product. A Tivoli upgrade image contains only the files that have changed since the previous product release, with the scripts and commands that are needed for installing the new files and configuring the database. Contrast with [Tivoli installation image](#).

**Tivoli User Administration.** A Tivoli product that provides a graphical user interface (GUI) for centralized management of user and group accounts. It offers efficient, automated management of user and system configuration parameters, secure delegation of administrative tasks, and centralized control of all user and group accounts in a network computing environment.

**Tivoli UserLink.** A Tivoli product that provides IP address synchronization between a PC agent and its associated PC managed node using the Dynamic Host Configuration Protocol (DHCP). Tivoli UserLink also enables a PC user to pull a file package to a Windows, Windows 95, or Windows NT workstation.

**TLL.** See [Task Library Language](#).

**TMB.** See [Tivoli Module Builder](#).

**TMD.** See [Tivoli Module Designer](#).

**TME 10™.** See [Tivoli Enterprise software](#).

**TMR.** See [Tivoli Management Region](#).

**TMR client.** In a Tivoli environment, any computer—except the TMR server—on which the Tivoli Management Framework is installed. The oserv daemon runs on the TMR client, and the TMR client maintains a local object database. See [Tivoli client](#) and [Tivoli server](#).

**TMR server.** A Tivoli server for a specific Tivoli Management Region (TMR). See [Tivoli client](#) and [TMR client](#).

**TNWR.** See [Tivoli NetWare repeater](#).

**toggle button.** In the AIXwindows Toolkit and the Enhanced X-Windows Toolkit, a graphical object that simulates a toggle switch; it switches sequentially from one optional state to another.

**tool palette.** In Tivoli NetView, a component of the graphical user interface (GUI) that enables the network operator to open application program instances by using the mouse to drag and drop the icons that represent the application program.

**topology.** In communications, the physical or logical arrangement of nodes in a network, especially the relationships among nodes and the links between them.

**topology console.** In the Tivoli Global Enterprise Manager and Tivoli NetView for OS/390, a Java-based graphical user interface that displays business system information from the topology server. The topology console displays each component as a separate icon or shape and draws lines between icons to denote links. It then uses color to indicate the status of each component and of the business system as a whole. As the topology server receives configuration and status updates for the business system, it updates the topology console. Therefore, the topology console always displays the real-time configuration and status of the business system.

**topology database.** See [local topology database](#) and [network topology database](#).

**topology database update (TDU).** A message about a new or changed link or node that is broadcast among APPN network nodes to maintain the network topology database, which is fully replicated in each network node. A TDU contains information that identifies the following:

- The sending node
- The node and link characteristics of various resources in the network
- The sequence number of the most recent update for each of the resources described.

**topology server.** In Tivoli Global Enterprise Manager and Tivoli NetView for OS/390, a server that interacts with instrumented applications in a business system and provides information for display on the topology console. The topology server receives heartbeat events from instrumented applications or components and determines the business system in which a component belongs. The topology server also queries instrumented applications for related applications and for the status of resources in the network.
its monitors. All of this information is used to create and maintain a view of each business system's configuration and availability on the topology console.

**trace.** A record of the execution of a computer program. It exhibits the sequences in which the instructions were executed. (A)

**transaction.** A specific set of input data that triggers execution of a specific process or job; a message destined for an application program.

**transit time.** See **response time.**

**Transmission Control Protocol/Internet Protocol (TCP/IP).** A set of communications protocols that support peer-to-peer connectivity functions for both local and wide area networks.

**trap.** In the Simple Network Management Protocol (SNMP), a message sent by a managed node (agent function) to a management station to report an exception condition.

**triggered response.** In a Tivoli environment, the action that is taken when a monitor reaches or exceeds a threshold.

**trouble ticket.** In Tivoli NetView, a record of a problem that has occurred. The trouble ticket becomes the formal vehicle to trace a problem from its occurrence to its resolution.

**TTL.** See **time to live.**

**tuple.** In a relational database, a part of a relation that uniquely describes an entity and its attribute. A tuple can be represented by one row of a relation table. (T)

**U**

**UDP.** See **User Datagram Protocol.**

**underlying connection.** In Tivoli NetView, the representation of lower-layer connectivity that is used by higher-layer connectivity. For example, the physical connection that transports data between two IP hosts is an underlying connection.

**unmarshall.** To copy data from a remote procedure call (RPC) packet. Stubs perform unmarshalling. Contrast with **marshall.**

**upcall.** In a Tivoli environment, a method invocation from an endpoint “up” to the gateway. Contrast with **downcall.**

**User Datagram Protocol (UDP).** In the Internet suite of protocols, a protocol that provides unreliable, connectionless datagram service. It enables an application program on one machine or process to send a datagram to an application program on another machine or process. UDP uses the Internet Protocol (IP) to deliver datagrams.

**user login map.** In a Tivoli environment, a mapping that associates a single user login name with a user account on a specified operating system. User login maps enable Tivoli administrators to log in to the Tivoli environment or perform operations within the Tivoli environment with a single user login name, regardless of the system that they are currently using.

**user plane.** In Tivoli NetView, the submap layer on which symbols of objects that are not managed by an application program are displayed. Symbols on the user plane are displayed with a shadow, which makes them appear higher than symbols on the application plane. See **background plane.**

**user role.** See **authorization role.**

**using node.** The NCP in the host's domain that reports a link error condition.

**V**

**validation.** The checking of data for correctness or for compliance with applicable standards, rules, and conventions. (A)

**validation policy.** In a Tivoli environment, policy that ensures that all resources in a policy region comply with the region’s established policy. Validation policy prevents Tivoli administrators from creating or modifying resources that do not conform to the policy of the policy region in which the resources were created.

**variable.** (1) In programming languages, a language object that may take different values, one at a time. The values of a variable are usually restricted to a certain data type. (I) (2) A quantity that can assume any of a given set of values. (A) (3) A name used to represent a data item whose value can be changed while the program is running. (4) In the Simple Network Management Protocol (SNMP), a match of an object instance name with an associated value. (5) In the NetView command list language, a character string beginning with “&” that is coded in a command list and is assigned a value during execution of the command list.
verb. (1) In Tivoli NetView for OS/390, the first word of a NetView command that is delimited by a blank or a comma and that indicates what action is to be taken. (2) See LU 6.2 verb.

view administrator. The part of the NetView Graphic Monitor Facility that downloads the views created by the view preprocessor and that provides these views to the graphic data server.

viewing filter. In Tivoli NetView for OS/390, the function that allows a user to select the alert data to be displayed on a terminal. All other stored data is blocked.

view manager. In the NetView Graphic Monitor Facility, a facility that generates views according to Resource Object Data Manager (RODM) definitions and that provides status changes to the graphic data server.

view preprocessor. The part of the NetView Graphic Monitor Facility that creates unformatted views of SNA resources from the VTAM definition library (VTAMLST).

view preprocessor resource. An SNA subarea resource whose status is reported by the resource status manager and is stored in the graphic data server (GDS) databases when views containing the resource are downloaded.

Virtual Telecommunications Access Method (VTAM). An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability.

vital product data (VPD). Information that uniquely defines system, hardware, software, and microcode elements of a processing system.

VPD. See vital product data.

VTAM. See Virtual Telecommunications Access Method.

W

webmaster. The person who is ultimately responsible for managing and maintaining a particular Web site.

well-behaved application program. An application program that runs without disruption to the network.

widget. (1) In the AIX operating system, a graphic device that can receive input from the keyboard or mouse and can communicate with an application or with another widget by means of a callback. Every widget is a member of only one class and always has a window associated with it. (2) The fundamental data type of the Enhanced X-Windows Toolkit. (3) An object that provides a user-interface abstraction; for example, a Scrollbar widget. It is the combination of an Enhanced X-Windows window (or subwindow) and its associated semantics. A widget implements procedures through its widget class structure.

wildcard character. See pattern-matching character.

with-request. A Printing Systems Manager (PSM) document transfer method in which the client transfers documents directly to the server. This is the default transfer method. Contrast with dce-pipe-pull.

wizard. A dialog within an application that uses step-by-step instructions to guide a user through a specific task.

working directory. The directory that is currently in use by an operating system or application. If no path is specified, this is the directory to which data is written, from which data is deleted, or in which data is searched.

work space. (1) That portion of main storage that is used by a computer program for temporary storage of data. (2) In Tivoli NetView, a container for a set of event cards that meet certain criteria. See event filter.

wrap count. In Tivoli NetView for OS/390, the number of events that can be retained in the database for a specific resource or the number of alerts that are retained in the database.

X

XCF. See cross-system coupling facility.

X Window System. A software system, developed by the Massachusetts Institute of Technology, that enables the user of a display to concurrently use multiple application programs through different windows of the display. The application programs may execute on different computers.

Y

Year 2000 challenge. A term used especially by the computer industry to refer to the problems, challenges, and issues involved in preparing computer systems and applications for transition to, and operation in, the twenty-first century. For example, many computer systems and applications use two digits to represent the year ("97" rather than 1997). When these computer systems and applications encounter the digits "00" for the year 2000, they can misinterpret this to mean the year 1900 and can produce computing errors or fail to function. Although some systems and applications may not be affected until the eve of the new millennium (on 31 December 1999), many systems and applications that use future dates (such as expiration dates for credit cards) have already experienced Year 2000 problems. This problem could also affect such things as elevator
controls; household appliances such as VCRs and programmable coffee makers; heating, cooling, and security systems; telephone calls; driver’s licenses; automated teller machines and bank vaults; and airline flight schedules.

**Year 2000 ready.** A product is Year 2000 ready if the product, when used in accordance with its associated documentation, is capable of correctly processing, providing, and/or receiving date data within and between the twentieth and twenty-first centuries, provided that all products (for example, hardware, software, and firmware) used with the product properly exchange accurate date data with it.

**Y2K.** See [Year 2000 challenge](#).

**Z**

**zombie process.** In the UNIX operating system, a process that has been terminated but has not been cleaned up by its parent process. The existence of a large number of zombie processes could indicate an errant network daemon or application. Zombie processes are sometimes called “lingering terminated processes.”

**zoom.** In a user interface, to progressively increase or decrease the size of a part of an image on a screen or in a window.

**Numerics**

**4700 Support Facility.** In Tivoli NetView for OS/390, a component that enables the monitoring and control of IBM 3600 and 4700 Finance Communication Systems. The 4700 Support Facility can record, analyze, and display performance and status data on IBM 3600 and 4700 Finance Communication Systems.
# Index

## A

about this book ix  
accessibility information xi  
accessing NetView, Web browser 27  
acknowledging, unacknowledging objects 32  
actions, events 66  
adding connections 48  
adding objects and symbols 31  
algorithms, layout 13, 41, 46  
application, nvsniffer 95  
application alert events 101  
application plane 48  
application properties 41  
applications, integrating with NetView 43, 50  
ARP cache information 91, 104  
authentication failure  
  information about 80  
  systems causing, list of 80  
autodiscovery, performance 121  
avtomatic configuration 98  
avtomatic layout 47  
avtomatic network discovery 85  
avtomatic re-Invocation, nvsniffer 97  
Availability of Nodes, Tracking 84  
avoiding problems with NetView 134

## B

background  
  graphics 43, 46, 47  
  plane 47  
backups, file 136  
boolean logic 34  
browsing MIBs 76  
building MIB applications 80

## C

cache, ARP 91, 104  
capability fields 31  
changing display presentation 24  
checking background processes 114  
checking configured interfaces 91  
checking daemon status 114  
checking event display 101  
checklists  
  agent MIB problems 133  
  hardware and software 135  
  installation 114  
  netmon daemon problems 130  
  ovstart, ovstop, and ovstatus 135  
  ovtopmd daemon problems 127  
  owvdb daemon problems 128  
  remote operation 136  
  snmpCollect daemon problems 133  
  solving an incorrect subnet mask 120  
  trapd daemon problems 132  
child submap  
  area 18  
  creating 45  
Cisco routers, discovering 87  
Cisco Routers Are Not Discovered, Unnumbered Serial Links on 124  
classifying network problems 102  
client/server, understanding 29  
client/server environment, managing maps 41  
client/server map propagation 30  
collecting MIB data 81  
collections, defining  
  description 32  
commands  
  \`\textbackslash usrnOV\textbackslash bin\textbackslash event -l \textbackslash` 62  
  ftp 103  
  Get 74  
  netmon 110  
  ovaddobj 116  
  ovdelobj 116  
  ovmapsnap 99  
  ovobjprint 34  
  ovstart 115  
  ovstatus 114, 115  
  netstat 118, 119, 123  
  Set 74  
community names 136  
components of network discovery 118  
configuration  
  displaying IP addresses 90  
  displaying services 92  
  events 65  
  graphical maps 53  
  information from nodes 85  
  interfaces 91  
  MIB information, retrieving 92  
  networks 85, 90  
  problems 114  
  SNMP nodes 94  
  time-out intervals 95  
configuration, automatic 98  
configuration, managing network 98  
configuration, manual 98  
connections  
  adding 48  
  multiple 48  
connectivity problems  
  across networks 105  
  errors 102  
  gathering information about 104  
  symptoms of 103  
  within same network 104  
console, NetView 15  
container objects 53  
creating submaps  
  child 45  
  independent 45  
critical objects display 102

189
critical propagated status 39
customizing
  maps 42
  NetView Graph tool 82
cutting, copying, pasting objects and symbols 40

d
daemon names
  netmon 118, 126, 128
  ovspsdm 126, 131
  ovtopmd 126
  ovwdb 126, 128
  snmpCollect 132, 133
  trapd 132
daemons
  event and trap processing 132
  not running 116
  reporting 116
  status 114
  topology, discovery, and database 126
  tracing 133
data collection, SNMP 140
data collection directory, maintaining 136
database, events, Microsoft access 140
default propagated status 38
defining collections
  description 32
defining symbol properties 35
Deinstallation 7
determining problem source 99
device errors 102
diagnosing
  network connectivity problems
    across networks 105
    task example 104, 106
    within same network 104
  network performance problems
    symptoms 107
    task example 107
  network service problems
    symptoms 110
    task example 111
dialog boxes, changing tables in 27
disability information xi
Discovering Cisco Routers 87
Discovering Services on Nodes 95
discovering the network 85
Discovery, status checking 97
displaying
  critical objects 102
  current events 63
  device information 102
  MIB interface information 92
  MIB system information 92
  object database 34
  working with displayed events 64
  displaying, hiding graph grids 84

e
edit menu options 16
  elements of windows
    child submap area 18
    new objects area 18
    status line 19
    submap stack 17
    the menu bar 16
  Enterprise Adapter, Tivoli Enterprise Console 3
  enterprise console integration 3
  error events 101
  error messages at startup 116
  event and trap processing daemons 132
  Event Browser
    customizing 64
    displaying events 63
    overview 61
    pausing 64
    starting 63
    working with displayed events 64
  event correlation 67
  event history, displaying 64
  event log, viewing 65
  events
    configuring 65
    defining 61
    displaying 63
    event history 64
    filtering 64
    viewing event log 65
    working with 61, 64, 101

f
figures
  browse MIB window 76
  connectivity, same network 105
  connectivity across networks 106
  critical status is propagated 39
  customized internet submap 56
  customized submap 56
  diagnosing network performance problems 108
  diagnosing network service problems 111
  graph tool 82
  line properties dialog box 83
  marginal status is propagated 39
  metaconnection submap 48
  MIB
    data collection 81
    loader window 76
    tree structure 75
    navigation tree window 24
    netmon daemon's network monitoring interactions 129
    NetView toolbar icons 20
    NetView windows at startup 9
    personal computer symbol 35
    submap window 15
    symbol on the user plane 48
map configuration, graphical
container objects 53
customized internet submaps 56
defining 53
map regions 53
map discovery, failure 122
map discovery problems 122
map events log file, client/server 140
map features, central 29
map features, local 29
map object and database information 122
map regions
creating 53
defined 53
examples 56
maps and submaps
child submap area 18
customizing 42, 43
ing 40
learning about 40, 43
managing in client/server environment 41
metaconnections 48
multiple, reasons for creating 40, 42
open map 42
snaps 43
submap layout
defined 46
using algorithms 46
using automatic layout 47
using new objects area 47
xmap application 57
menu bar 16
menu types
cascade 25
greyed items 25
pull-down 16, 25
shortcut 25
working with 25
metaconnection submaps
behavior 49
defining 48
properties 49
MIB
browser 76, 109
data collection 81
displaying interface information 92
displaying system information 92
kinds 74
loading and unloading 75
naming conventions 74
retrieving configuration information 92
tool builder 80, 102
MIB applications
building, modifying, deleting 80
running within a map 80
Microsoft system management server, installing
NetView 145
monitor menu options 16
monitoring
Ethernet errors 78
Ethernet performance 78
monitoring (continued)
Ethernet traffic 78
interface traffic 78
network configuration 90
performance and traffic 109
real-time performance data 78
SNMP
authentication failures 80
errors 79
network activity 79
operations 79
traffic 79
TCP connections 79
the network 101
N
name, changing client address 139
name, changing server address 139
naming conventions x
navigation tree window 23
netmon
checklist 130
daemon 118, 119, 126, 128
if netmon discovers no nodes beyond the router 124
if netmon identifies an SNMP node as Non-SNMP 125
NetView, additional tools 57
NetView client/server, model 29
NetView console 115
NetView log files, maintaining 139
NetView program
console 115
daemons 9
errors at startup 1, 10, 116
getting acquainted 13
installing 1
message log 10
navigation tree window 23
runtime problems 114
setup 3, 115
starting 9
stopping 117
submap window 14
toolbars 20
using 31
using windows 14
NetView setting, changing 3
NetView setup options
delete daemon from ovsuf startup file 116
dump data collected by snmpCollect daemon 123
manage at entries for log and trace files 136
monitor..events 137
object..new 122
paging space 133, 135
select daemons to stop or restart
event and trap processing daemons 132
NetView setup options (continued)
select daemons to stop or restart (continued)
procedure 115, 116, 131
topology, discovery and database daemons 126
set options for
daemon tracing 134
netmon 123
NetView daemons 116
verify and dump database 122
network aliases, using 90
network and subnetwork aliases 41
network configuration
displaying IP addresses 90
monitoring 90
network discovery
automatic 85
in problem determination 118
using seed files 89
network management 73, 115
Network Neighborhood 1
Network Neighborhood, installing 1
network performance problems
diagnosing 107
monitoring performance and traffic 109
symptoms of 107
network status, color 36, 102, 103
network topology events 101
networks
classes 120
outside of the domain 121
new objects area 18, 47
node configuration events 101
nvnsniffer, Logging the Progress of 97
nvnsniffer application 95
nvnsniffer applications Never Complete, One or More 138
nvnsniffer.conf File, Managing 95
nvnsniffer Discover Mode 96
nvnsniffer Status Mode 97

O
object
acknowledging, unacknowledging 32
capabilities 31
copying 40
cutting 40
displaying object database 34
learning about 31
managing, unmanaging 32
new objects area 18
pasting 40
properties 31
object database 34, 40
object menu options 16
object placed, incorrect segment 124
objects, displayed 123
off loading discovery, remote MLMs 87
Off Periods, polling, scheduling 93
online help 25
options menu 16, 17
ovaddobj command 116
ovdelobj command 116
ovspmd daemon 126, 131
ovstart command 115
ovstatus command 115
ovtopmd daemon 126, 127
ovwdb daemon 126, 128

P
paging support 69
performance, gathering information 109
performance considerations 141
performance during autodiscovery 121
performance reports, generating 84
performing network fault management 99
permissions 137
ping 104
planes
application 48
background 47
defining 47
user 48
polling
setting intervals 93
the network 100
printing graphed data 82
printing tables, dialog boxes 27
problem determination
checking daemon status 114
checking NetView console 115
daemons not starting 116
installation problems 114
network variables 118
startup errors 10, 116
processes, checking 135
propagated status
default 38
defining 38, 41
propagate at threshold value 39
propagate most critical 39
properties, object 31
property tips 15
pull-down menus 16, 17, 25

R
refresh layout 47
registering
agents (using lrf files) 116
daemons 116
regulating traffic
netmon broadcast 110
NetView 109
Remote MLMs, Off Loading Discovery 87
remote operation problems 136
reporting on daemons 115
reports, generating performance 84
resolving network problems 99
retrieving MIB configuration information 92
root submap 40, 45
router fault isolation, understanding 36
routers
not showing up on the network 123
showing up as a host system 125
routers, discovered 123
routing tables, viewing information 91
rulesets 68
runtime problems 114, 137

S
scaling
submaps 46
y-axis on graphs 84
Scheduling Off Periods, Polling 93
seed files 89
sending SNMP traps 62
servers/clients, changing locale settings 140
services, displaying configuration 92
Services on Nodes, Discovering 95
settings, modifying 93
settings, TAPI 71
shortcut keys, keyboard xi
shortcut menus 25
Silent Installation to Facilitate Installation via SMS,
Creating a 145
SmartSets
defining 32
snapshots, taking 43, 99
SNMP
configuring nodes 94
nodes 114
snmpCollect daemon 132, 133
software
checklist 135
managing 135
verify version 137
solving problems 137
starting
Current Events application 63
daemons 9, 115
graph application 81
NetView 9
startup problems 10, 116, 137
status
colors 32, 36
daemons 115
events 101
propagation
at thresholds 39
default 38
most critical 39
status propagation, multiple interfaces 38
stopping
daemons 117
NetView 117
submap explorer 19
Submap Explorer 19
submap menu options 16
subnet mask, incorrect 120
support, paging 69
symbols
behavior 35
colors 36
executable 35
explodable 35
label 36
learning about 34
location on submap planes 35, 47
properties 35
status 36
type, classes and subclasses 35
symptoms
NetView problems 113
network problems
connection 103, 104
performance 107

T
tables
application toolbar icons 22
comparison of NetView MIB applications 77
container objects 53
daemon tracing 134
default status rule 38
default symbol status colors 36
in dialog box, changing 27
layout algorithms 46
local registration files 117
map database information 40
network discovery configuration information 85
shortcut menus 26
standard toolbar icons 20
submap properties 44
symbol types 54
trace mask values for netmon daemon 134
using a two-button mouse 13
taking snapshots 99
submap windows
learning about 14, 43
status line 19
submap stack 17
submaps
creating a child 45
creating an independent 45
defined 43
home 45
internet 51
navigating 43
network 52
node 52
presentation 46
segment 52
status line 19
with metaconnections 48
working with 44
zooming 46
subnet mask, incorrect 120
support, paging 69
symbols
behavior 35
colors 36
executable 35
explodable 35
label 36
learning about 34
location on submap planes 35, 47
properties 35
status 36
type, classes and subclasses 35
symptoms
NetView problems 113
network problems
connection 103, 104
performance 107

T
tables
application toolbar icons 22
comparison of NetView MIB applications 77
container objects 53
daemon tracing 134
default status rule 38
default symbol status colors 36
in dialog box, changing 27
layout algorithms 46
local registration files 117
map database information 40
network discovery configuration information 85
shortcut menus 26
standard toolbar icons 20
submap properties 44
symbol types 54
trace mask values for netmon daemon 134
using a two-button mouse 13
taking snapshots 99
