

# **HP Network Node Manager**

## **Managing Your Network with HP OpenView Network Node Manager**

**Windows, HP-UX, Solaris, and Linux operating systems**



**Manufacturing Part Number : n/a**

**July, 2004**

© Copyright 1993-2004 Hewlett-Packard Development Company, L.P.

---

## Legal Notices

### **Warranty.**

*Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.*

A copy of the specific warranty terms applicable to your Hewlett-Packard product can be obtained from your local Sales and Service Office.

### **Restricted Rights Legend.**

Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.

Hewlett-Packard Company  
United States of America

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

### **Copyright Notices.**

No part of this document may be copied, reproduced, or translated to another language without the prior written consent of Hewlett-Packard Company. The information contained in this material is subject to change without notice.

©Copyright 1993-2004 Hewlett-Packard Development Company, L.P.

Contains software from AirMedia, Inc.

© Copyright 1996 AirMedia, Inc.

### **Trademark Notices.**

Java™ is a U.S. trademark of Sun Microsystems, Inc.

Linux is a U.S. registered trademark of Linus Torvalds.

Microsoft® is a U.S. registered trademark of Microsoft Corporation.

Windows® and MS Windows® are U.S. registered trademarks of Microsoft Corporation.

Netscape™ and Netscape Navigator™ are U.S. trademarks of Netscape Communications Corporation.

Oracle® is a registered U.S. trademark of Oracle Corporation, Redwood City, California.

UNIX® is a registered trademark of The Open Group.



## 1. Network Management with NNM

The Network Management Challenge .....	21
How Network Node Manager Can Help .....	21
Moving to Proactive Network Management .....	22
Poll for Network Information .....	22
Watch for Events .....	23
Network Management Functions .....	24
Fault/Problem Management .....	25
Performance Management .....	27
Configuration and Change Management .....	28
Additional HP OpenView Product Information .....	29

## 2. How Does NNM Work?

Collecting Critical Information .....	33
SNMP's Network Management Model .....	33
Desktop Management Interface (DMI) .....	36
Automatic Discovery and Layout .....	38
How IP Discovery and Layout Work .....	38
How IPX Discovery and Layout Work .....	39
Interactions Between IPX Discovery and IP Discovery .....	40
Level 2 Discovery .....	41
Event Reduction Capabilities .....	43
NNM's Databases .....	44
Operational Databases .....	44
Data Warehouse .....	45

## 3. Resources Available While Learning NNM

Resources Included with NNM .....	49
NNM Online Help System .....	49
Tip of the Day .....	51
Manuals: Printed and Online .....	51
Release Notes .....	53
Reference Pages (Manpages) .....	54
The contrib Directory .....	56
The support Directory .....	57
White Papers .....	57
Resources and Services Available Through HP .....	58
HP Web Sites .....	58

---

# Contents

HP Consulting Service .....	58
HP Education .....	59
HP Product Support .....	59
OpenView Forum .....	59
Summary of HP OpenView Web Sites .....	60

## 4. Planning Your NNM Configuration

Budgeting Time for Planning and Configuration .....	63
Time for Planning .....	63
Time for Configuration .....	63
Taking Over NNM from Someone Else .....	64
Understanding the Network Information Needs of Your Organization .....	65
Who Needs What Information? .....	65
Work Sheet for Planning .....	67
Knowing Your Network .....	69
Deciding to Manage or Not to Manage Devices .....	69
Starting with a Well-Configured Network .....	72
Consistent IP Addressing Scheme .....	72
Consistent Name Resolution Scheme .....	73
SNMP Agents and Their MIBs Configured .....	75
DMI Service Providers and Their MIFs Configured .....	76

## 5. Initial Network Discovery: Options and Troubleshooting

Install Network Node Manager (if you have not already done so) .....	82
Installing IPX Transport Software (Windows only) .....	82
Establishing Universal Pathnames .....	82
Discover the Network... Let NNM Do It .....	86
Overview .....	86
Start the NNM Services (background processes) .....	89
Make Sure NNM's Services Are Successfully Running .....	89
Open NNM .....	90
Configure an Inventory Report .....	90
Let NNM Run Over Night to Populate the Maps .....	90
Verify the Accuracy of the Initial Discovery .....	92
Printing the Inventory Report .....	93
Viewing the Properties of Your Network Configuration .....	94

Finding Specific Devices on the Map . . . . .	96
Interactively Expand/Limit Your Management Domain . . . . .	100
Adding Networks to Your Management Domain . . . . .	100
Managing an Unmanaged Network Device. . . . .	101
Unmanaging a Managed Network Device. . . . .	101
Automatically Expand Your Management Domain . . . . .	102
Create a Seed File Specifying Multiple IP Networks to Manage. . . . .	103
Use loadhosts . . . . .	105
Expand the IPX Hop Count (Windows only). . . . .	106
Automatically Limit Your Management Domain . . . . .	107
Create a netmon.noDiscover File to Exclude Devices. . . . .	108
Use loadhosts . . . . .	110
Create a Discovery Filter Identifying Which Devices to Include . . . . .	111
Modify the oid_to_type File to Unmanage Devices by Type. . . . .	112
Limit the IPX Hop Count (Windows only). . . . .	113
Troubleshooting Discovery . . . . .	114
IP Discovery and Layout . . . . .	114
General IP Suggestions . . . . .	127
IPX Discovery and Layout (Windows only) . . . . .	136
Stop Everything and Start Discovery Over Again . . . . .	144

## **6. Preserve Your Sanity: Backup and Polling Configuration**

Backup/Restore to Protect Your Investment of Time . . . . .	149
How Does the Backup Work? . . . . .	150
Backing Up and Archiving All Critical Files. . . . .	151
Restoring All of NNM (procedures/options). . . . .	155
Restoring Part of NNM (procedures/options) . . . . .	157
Troubleshooting Information . . . . .	158
Custom Scripts . . . . .	161
Controlling the Amount of Traffic Generated by NNM . . . . .	163
Status Polling . . . . .	166
Configuration-Check Polling . . . . .	168
Connector Topology Polling . . . . .	169
New Node Discovery Polling (IP, Level-2, and IPX) . . . . .	169
Controlling Level-2 Device Discovery and Layout . . . . .	170
Secondary Failure Polling. . . . .	177
Fine-Tuning the Polling Services. . . . .	178
Watching NNM's Polling Queue. . . . .	179

---

# Contents

Running NNM Without Network Polling .....	180
---	-----

## 7. Map Making Fundamentals

Maps versus Submaps .....	185
Maps .....	185
Submaps .....	186
Understanding Objects versus Symbols .....	189
Objects .....	190
Symbols .....	193

## 8. Map Customization

Putting It All Together .....	198
Which Maps for Which Users .....	198
Copying the Default Map .....	201
Your Map Strategy .....	201
Controlling the Display of Devices Attached to Switches or Bridges .....	203
Turning On Star Configuration for Attached Devices .....	204
Turning Off Star Configuration for Attached Devices .....	205
Giving Your Network Symbols Meaningful Names .....	207
Turning Connection Labels on or off .....	209
Configuring Trunking and Meshing .....	211
Establishing Submap Persistence Settings .....	215
By Logical-level of Your Network's Structure .....	216
By Presence of Specific Devices .....	216
Controlling Which Devices Appear on the Map .....	218
Creating a Map Filter .....	218
Using the Hide Feature .....	219
Changing/Adding Object Attribute Fields .....	221
Changing the Value in an Attribute Field .....	222
Changing Symbol Type to Switch Attribute Sets .....	223
Adding Attribute Fields in the Object Database .....	224
Making the Maps Look Like Your World .....	226
Controlling the Placement of the Symbols on Your Map .....	227
Adding Your Own Submaps .....	230
Customizing the Internet Level of Your Network Map .....	233
Customizing Network-, Segment-, or Node-Level Submaps .....	242



Background Graphics . . . . .	246
Creating Your Own Map Symbols . . . . .	249
What Is a Symbol . . . . .	249
Specifying the Placement and Size of the Submap Windows . . . . .	252
Window Geometry . . . . .	252
Submap Overlay . . . . .	252
Miscellaneous Configuration Changes . . . . .	254
Windows Operating System . . . . .	254
UNIX Operating Systems . . . . .	255
Controlling Symbol Status . . . . .	256
Object Status Colors . . . . .	256
Compound Status . . . . .	260
Creating New NNM Features to Meet Your Team's Needs . . . . .	264
Adding to NNM's Menu Structure . . . . .	264
Creating Executable Symbols with Custom Behaviors . . . . .	265

## 9. Controlling Map Access

Establishing and Communicating a Process for Requesting Changes to the Map . .	272
Not Allowing Team Members to Make Changes . . . . .	272
Allowing Team Members to Make Changes . . . . .	272
Setting User Preferences Within NNM . . . . .	273
Using Command Line Startup Options . . . . .	275
Options . . . . .	275
Examples . . . . .	276
Using Operating-System Level File Permissions for the Map . . . . .	278
Windows: Setting Permissions . . . . .	279
UNIX: Setting Permissions . . . . .	280
Using the Context Feature to Control Menu Choices . . . . .	282
Using NNM's Predefined Contexts . . . . .	282
Creating Your Own Contexts . . . . .	283
Using ARF Files to Control Menu Choices . . . . .	285
ARF File Modifications Example . . . . .	291
Allowing Others to View NNM from Many Computers . . . . .	303
Remote Consoles . . . . .	303
NNM's Web Interface . . . . .	304
Microsoft Terminal Services . . . . .	304
Closing All Current Sessions . . . . .	305

---

# Contents

## 10. Keeping Up with Events on Your Network: Beyond the Maps

How NNM's Event System Works . . . . .	309
SNMPv1 Traps / SNMPv2c Traps and Informs . . . . .	311
DMI Events (Indications) . . . . .	313
Alarm Browser Overview . . . . .	315
Displaying Alarms . . . . .	316
Alarm Categories/Alarm Browser Windows . . . . .	317
Acknowledging Alarms . . . . .	321
Filtering Alarms . . . . .	322
Deleting Alarms . . . . .	324
NNM's Map and the Alarm Browser . . . . .	325
Specifying Additional Actions on Alarms . . . . .	326
Launching Specific Views or URLs from Alarms . . . . .	327
Configuring the Alarm Browser . . . . .	329
Controlling the Size of the Event Database . . . . .	330
Controlling the Size of the Alarm Browser's State File . . . . .	331
Controlling How Many Alarms to Delete Automatically . . . . .	332
Copy or Restore the Alarm Browser's State File . . . . .	334
Controlling How the Alarm Browser Looks . . . . .	335
Assigning Alarm Categories . . . . .	335

## 11. Event Reduction Capabilities: Getting to the Root Cause

NNM's Event Reduction Capabilities . . . . .	340
Correlation Concepts . . . . .	343
De-Duplication of Alarms . . . . .	347
ECS Correlations . . . . .	349
NNM's Built-In Correlations . . . . .	349
Command Line Control . . . . .	365
Correlation File Structure . . . . .	366
Troubleshooting . . . . .	366
Obtaining Additional Event Correlations . . . . .	367
Correlation Composer Correlators . . . . .	371
NNM's Built-In Correlators . . . . .	371
Correlator Fact Store Files . . . . .	380
Troubleshooting . . . . .	381
Creating Additional Correlators . . . . .	382

What is an SNMP Variable-Binding and How Do I Identify One? . . . . .	387
Look within the MIB File . . . . .	387
Check the Alarm Message Text in the Alarm Browser . . . . .	388
Special SNMP Variable-Bindings within NNM Interface and Node Status Alarms	388

## 12. Customizing Events: Doing It Your Way

SNMP MIB Browser . . . . .	394
Loading MIBs in the MIB Database . . . . .	395
Prerequisites . . . . .	396
Procedure Tips . . . . .	397
DMI Browser . . . . .	399
Creating DMI Queries . . . . .	399
Loading DMI-to-SNMP Event Mappings . . . . .	401
If a MIB Translation of the MIF File Is Available . . . . .	402
If No MIB Translation of the MIF File Is Available . . . . .	402
Event Configuration Overview . . . . .	404
Prerequisites . . . . .	404
Event Configuration Window . . . . .	405
Controlling Alarm Message Posting and Text . . . . .	413
Defining Automatic Actions for Events . . . . .	415
Variables and Special Characters Allowed . . . . .	418
Defining Additional Actions . . . . .	423
Using the MIB Application Builder . . . . .	425
Prerequisites . . . . .	426
Procedure Tips . . . . .	427
Data Collection & Thresholds . . . . .	429
Prerequisites . . . . .	429
Procedure Tips . . . . .	431
Defining Thresholds for Monitored MIB Objects . . . . .	435
Collecting and Storing Textual Data . . . . .	442
Creating and Using MIB Expressions . . . . .	442
Unique Properties of the SNMP MIB Object sysObjectID . . . . .	447

## 13. Using Event Data

Graphing SNMP Data . . . . .	453
. . . . .	453
Graphing Historical and Real-Time Data in the Same Graph . . . . .	455
Grapher Operations . . . . .	456

---

# Contents

Printing a Graph .....	457
Data Warehouse .....	458
Data in the Data Warehouse .....	460
Reporting .....	460

## 14. NNM on the Web

Overview of Dynamic Views and Home Base .....	466
Dynamic Views .....	467
Modifying Dynamic View Menus .....	473
Overview of the Java-based Web Interface .....	474
Setting Up the HP OpenView Web .....	475
Role Configuration Files for the HP OpenView Web .....	476
The HP OpenView Launcher .....	482
Launcher User Interface .....	482
Configuring the Launcher .....	484
HP OpenView Network Presenter .....	491
Starting the Network Presenter .....	491
The Network Presenter Window .....	492
Network Presenter versus NNM on a Management Station .....	493
Configuring the Network Presenter .....	495
Using Symbol Registration Files and Bitmaps .....	496
Listing Multiple Maps in the Launcher .....	498
SNMP Data Presenter .....	499
Configuring the SNMP Data Presenter .....	501
Alarm Browser .....	502
Using and Configuring the Alarm Browser .....	503
Event Reduction .....	505
Reporting Interface .....	506

## 15. Maintaining NNM

Daily Tasks .....	511
Check Running Services .....	511
Check Disk Space .....	511
Trim Files .....	512
Weekly Tasks .....	515
Back Up NNM .....	515

Check Polling Performance . . . . .	515
Check Web Launcher Log Files . . . . .	516
Monthly Maintenance Tasks . . . . .	517
Check Patch Releases . . . . .	517
Check License Requirements . . . . .	517
Yearly Maintenance Tasks . . . . .	518
Evaluate Latest Release . . . . .	518
Other Maintenance Tasks . . . . .	519
Enabling/Disabling Automatic Startup . . . . .	519
Customizing the Startup Configuration . . . . .	520

## A. NNM Services and Files

Services and Files . . . . .	525
Background Services . . . . .	525
Foreground Services . . . . .	527
Web CGI Programs . . . . .	529
Behavior of the Services and Files . . . . .	531

## B. Troubleshooting NNM Itself

General Troubleshooting Considerations . . . . .	539
When You Need More Information . . . . .	539
Preventive Practices . . . . .	539
Characterizing the Problem . . . . .	542
Distinguish Local versus Remote Problems . . . . .	542
Remote Nodes' Level of Manageability . . . . .	542
Context: What Changed? . . . . .	543
Duration: How Long or How Often? . . . . .	543
Context: What Action Was Performed? . . . . .	543
Troubleshooting Background Services . . . . .	544
Associating Background Service Names and Port Numbers . . . . .	546
The httpd Background Service . . . . .	548
The netmon Background Service . . . . .	548
The ovactiond Background Service . . . . .	552
The ovalarmsrv Background Service . . . . .	552
The ovcapsd Background Service . . . . .	553
The ovspmd Background Service . . . . .	554
The ovrepld Background Service . . . . .	556
The ovtopmd Background Service . . . . .	556

---

# Contents

The ovtrapd Background Service .....	561
The ovuispmd Background Service .....	562
The ovwdb Background Service .....	564
The pmd Background Service .....	566
The ovrequestd Background Service .....	569
Troubleshooting Data Collector Problems .....	570
Set-community Name and Trap-dest Loss Upon Re-Installation .....	574
Troubleshooting Web Components .....	576
Language Selection Problems .....	577
HP OpenView Launcher .....	578
Network Presenter .....	581
Alarm Browser .....	582
SNMP Data Presenter .....	583
Troubleshooting NNM Operations .....	586
Runtime Components .....	586
Network Management Operations .....	589
Browsing an Internet MIB .....	591
Building and Executing MIB Applications .....	591
Configuring Events .....	594
Event Reduction Capabilities .....	595
Loading Internet MIBs .....	596
X Windows Components (UNIX operating systems only) .....	598
Online Help .....	599
Troubleshooting Windows Applications .....	600
Problems Launching Applications .....	600
Windows Operating System Tools .....	600
Recommended Logging and Tracing Practices .....	602
Logging and Tracing .....	602
Web Launcher Error Log .....	616
Improving Traffic Management and Performance .....	617
Traffic Management .....	617
Performance .....	619

## C. Changing All the Symbols for a Particular Device

Procedure .....	624
Example .....	625

Create a Symbol Registration File . . . . .	626
Copy and Modify the Symbol Graphics . . . . .	626
Map a sysObjectID to the New Symbol . . . . .	628
Define the Capabilities for Computer_workstation_700 . . . . .	629
Inform NNM about the New Fields . . . . .	630
Provide Additional Information in the oid_to_type File . . . . .	631
Update the Database . . . . .	632
Verify Symbol Changes . . . . .	633
Back Up Your Efforts . . . . .	635

## **D. Changing an Object's Vendor and SNMP Agent**

How NNM Discovers Vendor and SNMP Agent Values . . . . .	638
When the Vendor and SNMP Agent Would Not Be Set . . . . .	639
Procedure . . . . .	640
Example . . . . .	641

## **E. Reducing NNM's DNS Lookups**

Reducing Unresolved Hostname Lookups in NNM . . . . .	650
Reducing Unresolved Hostname Lookups in NNM with the No Lookup Cache . . . . .	652
Reducing IP to Hostname Lookups with the ipNoLookup.conf File . . . . .	654
Configuring and Enabling the ipNoLookup.conf File . . . . .	654
Reducing IP to Hostname Lookups with Negative IP Lookup Caching . . . . .	655
Using NNM's DNS Tracing Tool . . . . .	657
Enabling the DNS Tracing Tool . . . . .	657
The DNS Tracing Tool Log Files . . . . .	659
Using the resolveNames.ovpl Script . . . . .	662

<b>Index . . . . .</b>	<b>663</b>
------------------------	------------

---

# Contents



---

## Support

Please visit the HP OpenView web site at:

<http://openview.hp.com/>

There you will find contact information and details about the products, services, and support that HP OpenView offers.

You can go directly to the HP OpenView support web site at:

<http://support.openview.hp.com/>

The support site includes:

- Downloadable documentation
- Troubleshooting information
- Patches and updates
- Problem reporting
- Training information
- Support program information



---

# **1 Network Management with NNM**

This chapter describes the basic challenges of network management, and how Network Node Manager and other HP OpenView products address those challenges.

Topics covered include:

- How NNM can help you move toward proactive network management (page 22).
- An overview of each function of network management, and how NNM and other HP OpenView products address several of those functions (page 24).

Prior to reading this manual or using this product, you should have a good working knowledge of network management protocols such as the Simple Network Management Protocol (SNMP) and the Open Systems Interconnection (OSI) reference model. You should also understand the various types of network equipment and understand their functional differences.

## The Network Management Challenge

As a network administrator, you are under increasing pressure to perform the necessary tasks to meet the business demands of your organization. Some of the challenges you often face include:

- Dealing with user dissatisfaction in the face of increasing user expectations. Users want to be able to have the tools they want, when they want them to maintain their productivity.
- Deploying network services so that end users maintain productivity from their desktop systems.
- Keeping the systems operating 24 hours a day, 7 days a week.
- Producing faster, more predictable response times.
- Integrating and managing diverse and complex technologies — networks, servers, databases, applications, legacy systems, web-based systems—all from different vendors.
- Managing multiple sites and global distribution.
- Moving to proactive management techniques to keep it all running smoothly.

Clearly, the demands on a network administrator are great. Network management is now a high-pressure situation where more services are requested on tight schedules with fewer resources.

### How Network Node Manager Can Help

Network Node Manager is the foundation from which most of the HP OpenView products operate. When installed, the other HP OpenView products appear as added functionality within NNM. NNM not only functions as a solution on its own, but can collect data for, and forward data to, other HP OpenView products. It is the starting point for implementing a network management solution.

## Moving to Proactive Network Management

By using proactive, rather than reactive, network management, you can reap the true benefits of NNM. Several things must be present in order for proactive network management to be a reality.

- First, you must find out the current state of your network—what devices are present, how they are configured, how they are behaving, their performance levels, what is currently going wrong. NNM does this for you by polling for network information and screening problems through event correlation.
- Second, you must identify trends and determine how to optimize the network by changing configurations, replacing network devices, etc. NNM assists you by collecting historical information. You can easily access the historical data that NNM gathers and use it in your favorite statistical analysis programs.
- Third, you must learn how to predict what might go wrong, determine how to prevent it from happening, and learn to avoid future problems. NNM allows you to monitor thresholds of your choosing for critical network devices.

### Poll for Network Information

The first step to proactive network management is to gather all the information that you can about the current state of your network.

NNM continually polls for:

- Status of objects (for example, up, down, or marginal).
- Network topology changes, including the discovery of new nodes.
- Configuration changes.

You can configure thresholds for NNM to monitor on critical network devices and configure NNM to notify you or your team members when anything goes wrong. For example, you can set thresholds on CPU loads, disk space used, interface and link errors, and collected MIB data. You can configure NNM to dial a pager when any of these situations become critical.

## Watch for Events

NNM actively notifies you when an important event occurs. The event is reflected by a change in the color of the device's symbol on the map, and is reported through NNM's Alarm Browser.

Each category of alarms has a corresponding button that changes color to show the highest severity alarm in the category. Thus, you are immediately notified of potential or current problems.

You can browse through alarms to help you diagnose problems. You can view alarms by specific node or other sort criteria to help you solve problems quickly. The event reduction features in NNM monitor incoming alarms, identify patterns of common network problems, and for an identified problem, post one meaningful alarm with all related alarms nested beneath.

You can define the format of alarm messages so that they are most useful to your team. Your team can use the Alarm Browser to notify everyone else about which issues are being covered.

You can use event customization to automate some of the common fault management tasks for your team by using NNM to configure automatic actions upon the receipt of specific alarms. You can also filter out less important events, or simply send them to a log file.

## Network Management Functions

Network management can be defined in general terms as the ability to have a single point of control to accomplish the activities required to manage a network. Network Node Manager provides an integrated tool for the network manager to control and manage multiple networked systems and applications from a single graphical representation of the network.

When properly implemented, a network management station can result in:

- Reduced downtime of network systems and devices.
- Quick detection and correction of network problems without disrupting the network.
- The ability to monitor data to anticipate problems (predictive).
- The ability to log information for historical analysis.
- The ability to perform an action when some pre-defined event or situation has occurred.

To understand how Network Node Manager and other HP OpenView products address these problems, it is helpful to break network management down into functional categories, listed below. Following this list is an explanation of how NNM and other HP OpenView products help you manage these areas.

- **Fault and Problem Management**  
This function detects, isolates, and controls problems, or faults, on the network. This is carried out by network status monitoring, alarms, alerts, reporting, and predictive tools.
- **Performance Management**  
This function measures the performance of network hardware, software, and media, such as throughput rate, percentage utilization, error rates, and response time, through the collection and analysis of data about the network.
- **Configuration and Change Management**



This function is responsible for finding and setting up the network devices that control the behavior of the network. This function also includes central control of configurations.

- **Accounting Management**

This function obtains statistical information on network use. You can collect and process data related to resource consumption on the network, track each individual and group's utilization of resources, and control access to the network for individuals and groups.

- **Security Management**

This function protects the network and its interconnections, systems, and network management information from unauthorized access, unauthorized use, and other harm.

## **Fault/Problem Management**

### **NNM's Capabilities**

Often times the most difficult management task is identifying the source of a problem when it occurs. NNM helps you identify problems and errors, recognize trends, and proactively avoid problems. NNM lets you:

- Automatically discover the IP and IPX nodes in your network.
- Automatically monitor your network's status through the map interface and the event browser interface.
- Manage any vendor device that supports the Simple Network Management Protocol (SNMP). NNM manages both standard and enterprise-specific Management Information Base (MIB) objects.
- Manage non-SNMP nodes that use IP or IPX protocols.
- Manage any vendor device that supports the Desktop Management Interface (DMI). NNM monitors both standard and enterprise-specific Management Information Format (MIF) objects.
- Include new enterprise-specific Internet MIBs in NNM's MIB. Once you have loaded the new MIB module on the management station, you can access any of the MIB objects defined in that MIB module.
- Build new MIB applications (without programming) for Internet-standard and enterprise-specific MIBs. Once you have built a MIB application, you can monitor objects through the menu bar.

- Define event thresholds for MIB objects; for example, an event can be generated when the disk usage on a particular device exceeds a limit.
- Define actions to be taken upon receipt of an SNMP trap.
- Manage event storms and improve the information content of events by suppressing unwanted, redundant events and adding new, more informative events, using NNM's event reduction strategies: de-duplication, ECS correlations and Correlation Composer correlators.
- Integrate your own fault detection applications into the menu bar to yield an integrated management solution.
- Diagnose network faults and performance problems from one location, and diagnose problems by looking at trends over time. This includes customizing and automating the monitoring of your network and the response of management stations to events.
- Launch Systems Management Server (SMS) Administrator, or view the SMS properties of a node.
- Launch Windows operating system applets such as the Event Viewer, Performance Monitor, and Registry Editor from within NNM on remote systems with the Windows operating system.

### **Other HP OpenView Products' Capabilities**

**NNM's Extended Topology Functionality** NNM's Extended Topology functionality augments NNM's Advanced Edition by discovering and displaying additional device connectivity information that you can use to diagnose network problems. It includes, but is not limited to the following features:

- Management of heterogeneous switched layer 2 environments and routed layer 3 environments.
- An enhanced web user interface with dynamic views.
- Using SPIs (Smart Plug-ins), provides views from protocols and technology running on top of your network, such as OSPF and HSRP.
- Launching targeted views from events for rapid problem resolution.
- Discover and monitor network domains that contain overlapping addresses from the private addressing space.

**HP OpenView Operations** IHP OpenView Operations is an event and problem management solution that allows you to identify, locate, correlate, and resolve system and network faults. It includes:

- A central management console for monitoring events.
- Intelligent agents that can be preconfigured to solve problems immediately without interacting with the management console.
- Management of workstations running UNIX operating systems, the Windows operating systems, Novell Netware, and legacy systems.
- Integration with other OpenView products.

## Performance Management

### NNM's Capabilities

Figuring out how to fine-tune your network's performance can be a real challenge. To assist in this task, NNM gathers and reports information on network performance, availability, inventory, and exceptions. NNM can forward information to other HP OpenView products that provide statistical analysis on the data.

The following is a brief summary of NNM information gathering features:

- Automatically monitor your network's status.
- Collect historical MIB information about MIB objects and MIF events, store the data for trend analysis, and graph collected data. This helps you determine performance trends.
- Automatically set thresholds based upon standard deviations of historically collected data.
- Automatically collect data and generate General Availability and General Inventory reports.
- Manage and modify reports using the NNM Report Configuration Interface.
- Manage information being collected and stored in the data warehouse.
- Integrate your own performance monitoring applications with the menu bar to yield an integrated management solution.

- Diagnose network faults and performance problems from one location, and diagnose problems by looking at trends over time. This includes customizing and automating the monitoring of your network and the response of management stations to events.

## **Configuration and Change Management**

### **NNM's Capabilities**

Keeping track of the devices on your network, their configurations, and their interrelationships can be time consuming. NNM helps you keep a record of this information, by allowing you to:

- Store critical configuration information from devices such as routers and switches.
- Track inventory of the devices on the network.
- Generate snapshots of your network before any departmental moves to speed reconstruction after the move.

## **Additional HP OpenView Product Information**

You can find additional information about other HP OpenView products and solutions at <http://openview.hp.com>.



---

## **2** **How Does NNM Work?**

When you set up NNM to run on a management station, your team can access NNM from the management station, from remote consoles, or from the worldwide web. All three forms of access tie into the same databases running on the management station. Therefore, all members of your team continuously see the most current information. NNM works for you in the following ways, as explained in this chapter:

- Maps out your network for you. The map symbols change color to indicate if something is wrong (page 38).
- Collects critical information about your network and maintains a current log of alarms that you need to know about (page 33).
- Correlates collected information to help you quickly determine the probable cause of problems (page 43).
- Maintains a relational database from which data can be exported for historical analysis using your choice of statistical programs (page 44).
- Provides easy access to report generation, see the NNM online help for more information. See also the *Reporting and Data Analysis* online manual. You can create reports using a set of predefined templates that are populated from the data warehouse.

The remainder of this book then leads you through the process of understanding, configuring, customizing, and troubleshooting NNM.

The Extended Topology functionality augments NNM's Advanced Edition by discovering and displaying additional device connectivity information that you can use to diagnose network problems. See the *Using Network Node Manager Extended Topology* manual for more information.



## Collecting Critical Information

NNM uses several protocols to maintain communication channels with each managed device on your network. These include:

- SNMPv1; SNMPv2
- TCP/IP
- IPX/DMI
- UDP
- ICMP
- ARP/RARP

Since SNMP is the primary protocol used for NNM's communications, an understanding of the SNMP management model will help you better understand how NNM works.

NNM also uses other, lower-level families of protocols (sometimes referred to as services), such as the ARPA family, the Berkeley family, and the NFS family. These protocols are used for functions such as file transfer, e-mail, or remote login.

### SNMP's Network Management Model

The network management model consists of a network management station, managed nodes with agents, and a network management protocol. NNM uses SNMP as its standard protocol, and uses this protocol to communicate over other protocols, such as TCP/IP, IPX, and UDP.

#### The Role of Manager and Agents

A manageable network consists of one or more network management stations (manager), a collection of SNMP agents, and network objects.

- A **manager** is an application that executes network management operations to monitor and control agent systems. The implementation of these network management operations is called the manager.

- An **SNMP agent**, which resides on a managed node, is an application that acts on behalf of an **object** to perform network management operations requested by the manager.
- An **object** is anything that will be managed, such as a host, gateway, terminal server, hub, bridge, application, or database.

A manager and an agent may exist on the same system. The manager and agent communicate using the Simple Network Management Protocol (**SNMP**). Network Node Manager supports both SNMP version 1 (SNMPv1) and Community-based SNMP version 2 (SNMPv2C). SNMP permits the following activities:

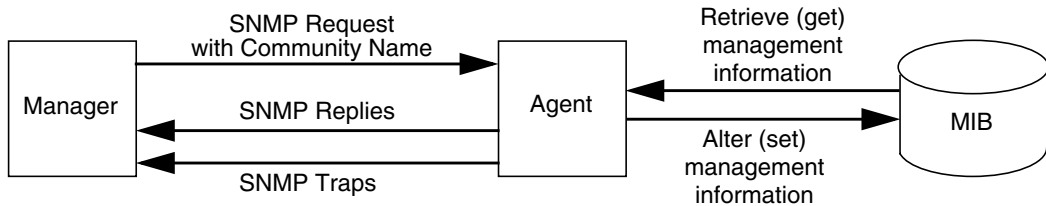
- A manager can retrieve management information from an agent. (The actual mechanism to retrieve management information depends upon the SNMP versions supported by the agent, and can include `get`, `getNext`, and `getBulk`.)
- A manager can alter, or `set`, management information on the agent's system.
- An agent can send information to the manager without an explicit request from the manager. This operation is called a **trap** in SNMPv1, or a **notification** in SNMPv2C.

Traps and notifications alert the manager to changes that occur on the agent system, such as a reboot. The agent must usually be preconfigured to know where to send traps or notifications (see page 311).

SNMP requests for information from an agent are accompanied by a **community name**. A community name is much like a password; it must be correctly received from the manager before information can be accessed from the agent. You must inform NNM about the community names that are in use on your network (see page 128). The manager accesses the agent's MIB object instances using SNMP's `get`, `getNext`, `getBulk`, and `set` operations.

Figure 2-1 shows a simplified diagram of the manager-agent interactions just described.

**Figure 2-1**      **Manager-Agent Communication through SNMP**



### Basic SNMP Services

There are six basic operations associated with SNMP. They are:

- get request. Reads a value from a specific variable.
- getNext request. Traverse information from a table of specific variables.
- getBulk request.
- get response. Replies to a get or a set request.
- set request. Writes a value into a specific variable.
- trap or notification. A message initiated by the agent without requiring the management station to send a request.

### Managed versus Unmanaged Objects

A **managed object** is one that is actively being polled by NNM to determine its status and configuration. An **unmanaged object** is one that exists in the databases and within maps, but is not being polled by NNM.

NNM gives you the choice of managing or not managing objects, depending on your information needs and network resources.

When an object is managed, NNM can obtain any information that you specify about that object (as long as that object's protocol is supported). The more objects you manage, the more memory and disk space is required on the management station. In addition to the information that you specify, the management station will need processing power for the

routine status and configuration polling and the event monitoring of each managed object. If an object is critical to the network function, then you should manage it.

If the object is not critical to the functioning of your network, you might choose to *unmanage* it; which means that NNM will not actively monitor the object. When an object is unmanaged, less memory and disk space and less processing time is required on the management station. However, you only get minimal information about that object on your map: its placement on the network and its IP/IPX address (static placement). NNM can still receive traps and post alarms in the Alarm Browser for unmanaged objects.

## Desktop Management Interface (DMI)

The Desktop Management Interface (DMI) is a parallel management strategy to SNMP. DMI standards were developed by the Desktop Management Task Force (DMTF). Although DMI is entirely independent from the SNMP protocol, there are many similarities:

- The DMI **client** is similar to the SNMP management station.
- The DMI **service provider** is similar to the SNMP agent and must be running on each remote DMI device.
- The DMI **MIF** (Management Information Format) file is similar to the SNMP MIB file and defines the management information that can be provided by the service provider (event) or requested by the client (get/set).
- The **DMI event** is similar to the SNMP trap.

If your management station is running NNM on the Windows operating system, during discovery polling, NNM **subscribes** to receive DMI events from all managed systems running DMI version 2.0. All DMI events are received by NNM's `ovcapsd` service (background process) and converted to SNMP traps as specified in the Desktop Management Task Force's DMI to SNMP Mapping Standard. The SNMP-equivalent message is posted to NNM's Alarm Browser.

---

### NOTE

If your management station is running NNM for a UNIX operating system, you must communicate with the DMI service provider indirectly through an SNMP agent installed on the remote DMI-enabled device itself along with a DMI-to-SNMP mapping agent. The SNMP agent then

handles all communications. Therefore, NNM doesn't need to subscribe to the DMI service provider in order to receive DMI events from this node.

---

## Automatic Discovery and Layout

When you start the NNM background processes, all IP and Level 2 devices (devices that support bridge, repeater, and MAU MIBs) on your network are automatically discovered and mapped out. If you are running NNM on a management station running the Windows operating system, IPX devices are also discovered and mapped out. This map is a visual representation of the communications channels established between NNM and the devices in your network. Be aware that this map is not a physical representation; rather, it is a logical representation. The accuracy of these communications channels between NNM and your network devices determine whether or not NNM can provide the information you need in order to manage your network.

The initial polling process may take several hours, or even over night, to discover all the devices on your network. However, it is worth the wait! You start benefiting from NNM immediately. You will use NNM's default map, and Alarm Browser to pin-point any gaps in your well-configured network scheme.

### How IP Discovery and Layout Work

The `netmon` service (background process) uses a combination of SNMP requests and ICMP pings transmitted over UDP and IPX to find out about the nodes on your network.

To discover the nodes on the network, `netmon` needs access to the following information:

- The subnet mask from the agent on the management station.
- The address of the default router in the management station's routing table.
- SNMP information from, at a minimum, the default router, and from other routers and nodes on the network.

For `netmon` to work, it requires the following:

- The management station must be correctly configured for networking.
- The management station must be running an SNMP agent.

- Nodes must be up and responding to ping requests to be discovered.
- All gateways/routers and the management station must have correctly configured subnet masks for all interfaces.

During IP discovery, `netmon` works best in these situations:

- The more routers in the network the better; and those routers should be running configured SNMP agents.
- The more nodes running configured SNMP agents the better.
- The more SNMP information `netmon` gets the better; for example, valid ARP caches and, from serial connections to routers, routing tables.

Information about the discovered nodes is stored in NNM's databases and is used to automatically generate the network map.

Over time, NNM will discover new nodes on the network. However, if a new node never talks with a gateway or other nodes on the network that support SNMP, NNM may not find it. In this instance you can use menu items within NNM to send a low-level ICMP ping that forces NNM to discover the node; or you can add the node manually.

## How IPX Discovery and Layout Work

If you are running NNM on a management station running the Windows operating system, in addition to discovering IP nodes on your network, `netmon` performs IPX discovery of your network. However, the methods used are very different from those used for IP discovery. The IP discovery process relies primarily on SNMP queries to gain hints of new IP nodes. For IPX discovery, `netmon` relies on using broadcast versions of various IPX protocols to discover the nodes, then uses SNMP once the node is discovered to fill in additional information about the node. In particular, `netmon` will be able to determine that an IPX node supports IP only if that node supports SNMP over IPX.

While the discovery model is different, the layout results in the submap are very similar. IPX network symbols are created in the Internet submap, connected by IPX gateways. IPX network symbols can be exploded to display segment and node submaps. However, IPX nodes always are connected to a single segment in the IPX network.

To discover IPX nodes on the network, `netmon` requires the following:

- The management station must be correctly configured for IPX networking.
- At least one IPX server or router must be connected to the same network as the management station.
- Nodes must be up and responding to IPX diagnostic requests to be discovered.

During IPX discovery, `netmon` works best in the following conditions:

- The management station is not also an IPX/NetWare gateway.
- Servers, routers, and nodes in the environment respond to IPX diagnostic, Routing Information Protocol (RIP), and Service Advertising Protocol (SAP) requests.
- IPX nodes in the environment support SNMP over IPX.

As with IP discovery, the information about the discovered nodes is stored in the NNM's databases and is used to automatically generate the network map.

Because `netmon` uses broadcasts, most IPX nodes should be recognized after the first few IPX polling cycles. After the default IPX discovery interval of six hours, most IPX nodes will be discovered, with the majority coming soon after the initial discovery cycle at system installation or startup. However, this discovery rate will vary depending on the number and type of nodes in the network. So before you start to troubleshoot `netmon`, allow the process sufficient time to discover your network. This may require allowing IPX discovery to run over night.

## Interactions Between IPX Discovery and IP Discovery

If an IPX node supports SNMP over IPX, `netmon` will attempt to use the MIB-II `ipAddrTable` to also include IP addresses and interfaces on the IPX node, as well as using the other MIB-II variables to provide additional information. When a node is discovered to support both IP and IPX, the following rules apply:

- Preference is generally given to the IP values as the name and labels for the node.
- After a node is determined to support IP, IP will be the preferred SNMP transport protocol.



- If a node is first discovered as an IP node, and a later IPX node is discovered to have the same set of IP addresses via SNMP, the IP node may be deleted in favor of the IPX node. For this reason, we recommend that you delay extensive map customization until IPX discovery is complete.

## Level 2 Discovery

NNM takes advantage of information provided in three standard MIBs to discover bridges (switches) and hubs. These MIBs are the bridge MIB (RFC 1493), the repeater MIB (RFC 2108), and the 802.3 MAU MIB (RFC 1515). If a network device supports any of these MIBs, `netmon` will use the information reported to develop a model of the topology which better represents how and to what the device is interconnected.

In the case of switches and bridges, additional information is gathered from the bridge MIB. In the case of hubs, additional information is gathered either from the repeater MIB or the 802.3MAU MIB.

The status of non-IP or non-IPX interfaces on switches, bridges, and hubs is determined via SNMP based on the administrative and operating status of the port.

If the MAC address of a node, which does not support IP or IPX, is reported by any of these MIBs, the node is added to the topology relative to the device which hears it. A label for this device is determined by the MAC address reported by the network interface card. Because no MAC layer control message protocol is available, the status of the devices is determined indirectly via information provided by the surrounding network elements.

NNM's Advanced Edition contains Extended Topology functionality. This additional functionality provides management of heterogeneous switched layer 2 environments and routed layer 3 environments. See the *Using Network Node Manager Extended Topology* manual for more information.

## Trunking and Meshing

Configuring two or more ports between two network devices into a single group is referred to as port trunking. This is done to increase the available bandwidth between the two devices. For example, you can

increase the available bandwidth between two devices, such as two ethernet switches, by configuring a group of two or more connections between the two devices.

Meshing involves connecting devices on the network via multiple paths. Meshing is used when devices are used in a critical function and redundant network connectivity is required.

You can configure NNM to label port trunking and meshing connections with information you add to the `netmon.equivPorts` file. See “Configuring Trunking and Meshing” on page 211 for more information.

## Event Reduction Capabilities

NNM includes several event reduction capabilities that identify relationships between different events to quickly determine the root cause of problems, such as event storms. This improves the information content of alarms by suppressing unwanted, redundant alarms and adding new, more meaningful alarms.

You see fewer, but more informative alarms that show the relationships and dependencies between network events. It then becomes easier to identify trends, to isolate important events, and to react more quickly to problems. You view these events through the NNM Alarm Browser.

Event reduction involves processing events based upon relationships between individual events. Event reduction:

- Analyzes events based upon previous, current, or subsequent events.
- Can create new events.

By analyzing the relationships between events in the stream, single events can replace groups of events. These events may have increased information content, and can be of a different protocol, compared to the original events.

- Dramatically reduces the number of alarms displayed by the NNM Alarm Browser.
- Correlates alarms between different event protocols.

NNM contains pre-defined event reduction configurations that you can modify, if desired. You can also create your own new ones to address specific needs within your environment. Refer to “Event Reduction Capabilities: Getting to the Root Cause” on page 339 for more information about these features.

## NNM's Databases

NNM provides several operational databases, each designed to store specific kinds of data and used for a variety of purposes. In addition, NNM includes a data warehouse: a relational database used to store historical information about your network.

For more information about the NNM databases and the data warehouse, consult the online manual *Reporting and Data Analysis with HP OpenView Network Node Manager*.

### Operational Databases

NNM's operational databases—object, map, topology, trend, and event—can be thought of as a single logical database. These databases store the operational data used by NNM, putting critical information at your fingertips.

#### The Object Database

The object database contains semantic information about symbols on the map. The information is generic; that is, it is not customized to any specific application. The object database contains field definitions such as `sysObjectID`, `vendor`, and `SNMP Agent`. You can see the field values in the object database for a particular object through the `Properties` dialog box for that object.

#### Map Database

The map database contains presentation information specific to each map. Examples of presentation information that is stored in the map database include the exact symbol placement on the map, the symbol associated with each object, and symbol labels. NNM updates the map database based on requests from the user, or from various NNM services.

#### Topology Database

The topology database manages information critical to the management of IP nodes. It includes state information, such as time stamps indicating when the object last changed and when it should be polled next. This

information helps NNM detect changes and communicate them to various NNM services. The topology database is stored in a proprietary HP OpenView format.

### **Event Database**

The event database is the repository for SNMP traps and OVW events that are received by NNM. It also stores events that are output from the Event Correlation Service (ECS). In addition, the event database stores information for the event browser tools about the state of the events displayed in the web browser applications. The information in the event database is displayed in the Alarm Browser.

### **Trend Database**

The trend database (sometimes called the `snmpCollect` database) stores MIB data and threshold information that is gathered through the `snmpCollect` service (process). Information from the trend database is used in the Reporting and the Data Collection features, and can be viewed with the Grapher feature of NNM on a management station.

### **Data Warehouse**

The NNM data warehouse stores data exported from the NNM databases into a relational database. You export this data to the data warehouse using the `Tools:Data Warehouse` menu choices (`Export Topology`, `Export Events`, and `Export Trend Data`).

You can use the information in the data warehouse to define your own reports. You can also use the NNM web Reporting interface to configure and view reports. When you create a report using the web Reporting interface, NNM automatically starts exporting the appropriate data to the data warehouse.

An embedded relational database is provided with NNM, and Oracle® and SQLServer are also supported.

How Does NNM Work?

**NNM's Databases**

---

# **3 Resources Available While Learning NNM**

There are many resources included with NNM that will help you use the product more effectively. In addition, there are also multiple additional services available through HP.

Resources included with NNM:

- Online help system (page 49)
- Tip of the Day (page 51)
- NNM's manuals, both printed and online (page 51)
- Release notes (page 53)
- Online reference pages/manpages (page 54)
- contrib programs (page 56)
- support directory (page 57)
- White papers (page 57)

Resources and services available through HP:

- HP web sites (page 58)
- HP Consulting Service (page 58)
- HP Education (page 59).
- HP Product Support (page 59)
- OpenView Forum (page 59)



## Resources Included with NNM

### NNM Online Help System

NNM includes an online help system that contains all the specific procedural information you will need while you are working with NNM. The help system has extensive search capabilities and a comprehensive index so you can easily find information. You can print or copy pages or sections of help information that are appropriate for the most common tasks for your organization.

---

**TIP**

The help system includes hyperlinks that make it more convenient and informative. By clicking on designated link words, you gain instant access to more topics and information related to your currently selected topic.

---

The online help system enables you to display information on the features of NNM, explains how to perform tasks with NNM, and includes conceptual information about how NNM works. Help entries are available from three general locations:

- Help menu
- [Help] button on dialog boxes
- Help from some symbol pop-up menus

### Using the Help Menu

The contents of the Help menu varies from one submap to another, depending on additional installed applications. The Help menu also differs between NNM on a workstation running the UNIX operating system, NNM on a workstation running the Windows operating system, and the NNM web interface.

## Printing Information from Online Help

NNM for the Windows operating system uses a standard Microsoft® Windows help system. Each help window has a `Print` button or menu item that you can use to print the current topic. You can also select multiple topics in the help table of contents, then print those topics.

On UNIX operating systems, you can select a topic in the `Contents` window of help, then select the `Print` menu item. In the `Print` dialog box, select `Print Topic` to print only that topic. If you want to print all the subtopics under that topic, select `Print Topic and All Subtopics`.

You can then combine these help printouts into a binder for future reference. This is especially useful for frequently-accessed information, so that you do not have to access the help system every time you need to look up a common task or function. You can also use these printouts to create your own operating manuals.

## Searching for Information

The most efficient way to find a particular help topic is index search. You access index search via the `Index` tab in the help system on a Windows operating system or by the `Index` button on a UNIX operating system. Every topic in the help system is indexed by the keywords that are used in the topic. When using index search, enter the word that best describes what you are looking for and that is most likely to narrow down your search. For example, if you want to find help on automatic map generation, enter the word “automatic” or “auto-generate” rather than the word “map.” Those words are more likely to find the entry that you want.

In a Windows operating system only: another way to find information is by using the `Find` function in the online help. The `Find` function searches every word in every topic for the word or phrase of your choice, and returns a list of the titles of topics where those words or phrases appear.

## Customizing Information for Your Organization

You can create custom information for your team or for a specific site within your organization. Simply copy and paste the contents of a help window to a word processor. Then add your own site-specific information or specific procedural information for your tasks to the help text. This can save you time when writing your own training or procedural manuals.

## Using Online Help Independently from the Software (Windows only)

The NNM help system on the Windows operating system is designed so that you can copy the help volume onto a diskette, take it with you, and run it independently of the software—whether on your home computer or traveling with you on your laptop.

Copy the following files to a floppy disk. These files are found in the `install_dir\help\%LANG%` directory:

- `help_file_name.hlp`
- `help_file_name.cnt`

Double-click on the `help_file_name.hlp` file to start the help system.

## Using Dialog Box Help

The [Help] button in a dialog box explains the purpose of the dialog box.

## Using Symbol Pop-up Menu Help

The right mouse button pop-up menu on a selected symbol may contain a help item for that symbol. You can customize this menu through the application registration file (ARF) to add a help topic if it does not already contain one.

## Tip of the Day

When you start NNM on the Windows operating system, the Tip of the Day appears. Add new tips by editing the `install_dir\help\%LANG%\tipOfTheDay.txt` file. Turn off Tip of the Day by removing the check from the Show tips at startup check box located in the Help:Tip of the Day menu.

## Manuals: Printed and Online

NNM includes a library of detailed reference materials for your use. Some are provided in printed form, some are provided online as PDF files. Get additional information from the online help or the release notes. The following manuals are provided with NNM.

- *Installation Guide for HP OpenView Network Node Manager*

This manual contains instructions about installing and configuring NNM.

- *Welcome to Network Node Manager*

An overview of NNM, its features, and its capabilities.

- *Managing Your Network with HP OpenView Network Node Manager*

This manual (that you are currently reading) contains detailed information to help network administrators configure, maintain, and troubleshoot NNM for their team.

- *Correlation Composer's Guide*

This manual provides information about the HP OpenView Correlation Composer. You can use the Correlation Composer to create new event correlation logic. Be sure to read the information in Chapter 11, “Event Reduction Capabilities: Getting to the Root Cause,” on page 339 before you start.

- *Guide to Scalability and Distribution for HP OpenView Network Node Manager*

This advanced manual outlines the steps you need to take to deploy NNM across multiple management stations within your organization. It contains specific configuration procedures for using the scaling and distribution features of NNM. These features are important when the size of your organization's network exceeds the resources of one NNM management station. They allow you to divide the work load among multiple management stations, yet share the network management information as needed.

- *Using Network Node Manager Extended Topology*

This manual explains how to use NNM's Extended Topology functionality to discover and display layer 2 device connectivity. It also explains how to use NNM's Extended Topology functionality and the Advanced Routing SPI (Smart Plug-in) to discover and manage networks that use IPv6, OSPF, and HSRP. This functionality comes with Network Node Manager's Advanced Edition. It is not included with Network Node Manager's Starter Edition.

- *Reporting and Data Analysis with HP OpenView Network Node Manager*

This manual is available only online and contains information about configuring and using NNM's reporting and data warehouse features. This book also contains information about how to export the historical data collected by NNM for your use with other statistical analysis programs of your choice.

- *Creating and Using Registration Files*

This advanced manual is available only online and contains information about how to create and modify NNM's registration files, customizing the appearance and behavior of NNM.

- *Integrating HP OpenView Reporter*

NNM and HP OpenView Reporter can be integrated, so that Reporter can query the NNM data warehouse to generate reports that help you to solve real network problems. This manual shows you how to configure the integration, as well as how to run and view reports. It also describes the preconfigured reports that are provided.

To access the online manuals shipped with NNM, use the **Help** menu. When you choose **Online Manuals**, a list of manuals appears. Choose the manual you want to view, and follow the instructions in the viewer.

Numerous additional HP OpenView online manuals are available at the following web site: <http://www.docs.hp.com>

## Release Notes

For information about NNM that becomes available after this book is published, refer to the **ReleaseNotes** directory or the **README.html** file on the CD-ROM.

Release notes are available on your hard drive after installing NNM. You can access the release notes from NNM's menu **Help:NNM:Release Notes** (or on the Windows operating system under **Start:Programs:HP OpenView:HP OpenView Online Documentation:NNM Release Notes**).

You can also launch the release notes in your web browser:

**Windows:** `install_dir\www\htdocs\%LANG%\ReleaseNotes\README.html`

**UNIX:** `/opt/OV/www\htdocs/$LANG/ReleaseNotes/README.html`

---

**NOTE**

In this manual, C and \$LANG in directory listings are variables for the language currently in use. C contains the English version.

---

## Reference Pages (Manpages)

Reference pages (Windows operating system) or manpages (UNIX operating system) document NNM configuration possibilities in much greater detail than you will find any place else. These pages include information about every possible parameter and available command line option, as well as useful background information that may be helpful when troubleshooting issues. These pages are named according to the function or feature that they are documenting.

Throughout this book, you will be referred to specific reference pages (manpages) for more information about the topic being discussed. You may want to browse through this resource collection yourself. You will find many more reference pages than are mentioned in this book.

### Accessing Reference Pages

On Windows operating systems, reference pages are provided through the help system. You can access them from these locations:

- Start:Programs->HP OpenView->HP OpenView Online Documentation->NNM Reference Pages.
- Any submap within NNM, through the Help menu.

---

**TIP**

If you are not sure which reference page to access for a specific issue, use the Find feature of the help system. Find searches every word in every reference page and provides a list of all places where the word appears. For example, if you want to learn about the netmon service's parameter choices and relationship to other services, type netmon in the Find dialog box.

---

## Accessing Manpages

On HP-UX and Linux systems, the following procedure is just one suggestion for displaying or printing manpages. This procedure requires that you have manpages installed locally on your system. (If your network provides manpages remotely instead (for example, from a central server), then check with your system administrator about how to access them.)

1. Determine where on your system the manpage files are kept. Type

```
echo $MANPATH
```

You should see a list with one or more directories. Multiple directories will be separated by colons (for example, `/usr/local/man:/usr/man`). It is recommended that you check the contents of each directory to make sure it actually has manpage files in it.

If you do not see a list of directories or you get no response to the `echo $MANPATH` command, refer to the section “Establishing Universal Pathnames” in Chapter 5.

2. Determine what syntax and options the `man` command is using. At a command-line prompt, type:

```
strings /usr/bin/man | grep col
```

You should receive a message similar to the following:

```
tbl -TX %s |neqn|nroff -h -man|col -x > %s  
tbl -TX %s |neqn|nroff -man|col -x|%s
```

3. Use the command syntax shown in Step 2 in one of two ways:

- Specify the qualified path of the directory containing manpage files as the `%s` value in the command.

Example: you know that the `bggen(1)` manpage is stored in the `/usr/man/man1` directory on your system, but you are not currently in that directory. To display this manpage online, type the following command:

```
tbl -TX /usr/man/man1/bggen.1 |neqn|nroff  
-man|col|more
```

Example: to send the `bggen(1)` command’s manpage to a file for printing instead, type the following:

```
tbl -TX /usr/man/man1/bggen.1 |neqn|nroff -man|col > /filename
```

- cd to the directory on your system that contains manpage files. Then specify just the command name and number as the %s value.

Example: you are in the /usr/man/man1 directory. Display the bggen(1) command's manpage online by typing:

```
tbl -TX bggen.1 |neqn|nroff -man|col| more
```

Example: to send this manpage to a file for printing instead, type the following:

```
tbl -TX bggen.1 |neqn|nroff -man|col > /filename
```

On Solaris systems, consult your system documentation for information on accessing manpages online or printing them.

## The contrib Directory

The NNM contrib directory is a collection of programs and extensions to existing NNM programs that HP programmers have developed and want to share with NNM users.

The contrib directory contains a README file explaining the files in the directory. Many of the individual programs have README files as well, which contain information specific to the program.

When you install NNM on a UNIX operating system, the contrib directory is installed by default.

To install the contrib directory on a Windows operating system, you must use the "Custom Installation" feature.

The files are installed in:

*Windows:* install\_dir\contrib\NNM

*UNIX:* \$OV\_CONTRIB/NNM

These programs are not supported by Hewlett-Packard and, as such, you use them at your own risk.



## The support Directory

The NNM `support` directory is a collection of programs and Perl scripts that HP support engineers have developed and want to share with NNM users.

The `support` directory contains a `README` file explaining the files in the directory.

When you install NNM on a UNIX operating system, the `support` directory is installed by default.

To install the `support` directory on the Windows operating system, you must copy and paste the `support` directory from the NNM Installation CD to the management station's hard drive. Place the `support` directory in the following location:

The files are installed in:

*Windows:* `install_dir\support\*.*`

*UNIX:* `opt/OV/support/*`

These programs are not supported by Hewlett-Packard and, as such, you use them at your own risk.

## White Papers

White papers are documents that explain various technical aspects of NNM and network management in general. Often these topics are not documented elsewhere.

When you install NNM on a UNIX operating system, the current white papers are installed by default.

To install the white papers on a Windows operating system, you must use the “Custom Installation” feature.

The white papers are installed in:

*Windows:* `install_dir\Doc\WhitePapers`

*UNIX:* `OV_DOC/WhitePapers`

## Resources and Services Available Through HP

You can find more information from the following sources:

- HP web sites
- HP Consulting Services
- HP Education
- HP Product Support
- OpenView Forum

### HP Web Sites

#### General Information

Using a web browser, you can learn more about HP OpenView by visiting the main HP OpenView web page at: <http://openview.hp.com>.

This web site contains information on current products, as well as literature that you can print or download for viewing. There are links to HP technical support and sales offices.

#### Product Manuals

There is also a web site for accessing current HP OpenView product technical manuals that you can download and print. These manuals are available in PDF format. Its URL is:

<http://openview.hp.com>

Click on the Support icon, then select Manuals.

Another web site presents HP OpenView product manuals in searchable web-based format: <http://www.docs.hp.com>.

### HP Consulting Service

The HP Consulting Service provides a wide range of consulting services, including initial installation and configuration, and network planning and organization. They can also help you plan and implement advanced strategies to help you take the best advantage of NNM's capabilities to

meet the specific needs of your organization. For more information about their services, visit the main HP Consulting web site, <http://www.hp.com/go/consulting>.

## **HP Education**

HP Education presents numerous classes on Network Node Manager and HP OpenView, geared toward different levels of training. For more information about course descriptions and class schedules, visit the main HP OpenView web site, <http://openview.hp.com>.

## **HP Product Support**

HP has more than 40 Response Centers worldwide offering telephone support in native languages and time zones. To register for telephone support, contact your local HP sales representative, or browse the HP OpenView web site at <http://openview.hp.com>.

In addition, HP offers software and materials support contracts so that you can automatically receive the newest versions of applications as they are available.

## **OpenView Forum**

OpenView Forum is an association of users and developers of HP OpenView network and system management solutions. An independent corporation, the OpenView Forum represents a vast body of practical knowledge and experience provided through conferences, a web server, an email reflector, and personal contacts.

The OpenView Forum online discussion center provides members with a place where they can gain information and knowledge, and provide feedback to software developers about HP OpenView products.

OpenView Forum sponsors a conference each year, which is attended by vendors and more than 1000 Forum members. You can find out more about OpenView Forum at their web site: <http://www.ovforum.org>.

## Summary of HP OpenView Web Sites

Table 3-1 presents a summary of the HP OpenView Web sites.

**Table 3-1** Summary of HP OpenView Web Sites

HP Web Site	Contents
openview.hp.com	The primary web site for HP OpenView. Contains links to: HP OpenView training information Product literature Product manuals
www.hp.com/go/consulting	HP Consulting Services
www.docs.hp.com	Product manuals in searchable web-based format.
openview.hp.com-click on support panel	Access to HP support.
www.ovforum.org	OpenView Forum's web site where you can exchange information with other OpenView users.

---

---

# 4

## **Planning Your NNM Configuration**

In order to get NNM the most benefit from NNM, you will probably want to configure NNM for your particular environment. NNM is a powerful program that can be configured in many ways. You need to budget time for planning how you wish to configure NNM and time for actually doing the configuration.

In order to make the whole process run as smoothly as possible, this chapter explains the issues to consider as you begin:

- “Budgeting Time for Planning and Configuration” on page 63
- “Taking Over NNM from Someone Else” on page 64
- “Understanding the Network Information Needs of Your Organization” on page 65
- “Knowing Your Network” on page 69
- “Starting with a Well-Configured Network” on page 72

## **Budgeting Time for Planning and Configuration**

NNM is an extremely flexible program that can be configured to meet a large number of business needs. Read through this entire chapter before you begin configuring NNM. Clearly identify the specific ways in which you will use NNM to assist you in managing your management domain, to minimize the time you spend configuring NNM.

### **Time for Planning**

If your management domain is small, this planning process could be quick. If your management domain is large and complex, it may take multiple meetings to gather the information that you need.

### **Time for Configuration**

A quick sketch of the configuration process outlined in this book is:

- Install NNM--initial discovery, customization, and troubleshooting.
- Establish backup procedures for NNM's critical files.
- Configure NNM's active polling configuration.
- Configure the maps.
- Configure the event monitoring system.
- Decide what historical information is most useful, and establish data collection toward that objective.
- Establish the maintenance procedures and schedule.

Network management is complex and specific to each organization's needs. NNM can best assist you to meet your network management goals if it is optimally configured. Allow sufficient of time for the configuration process.

You need to have a clear idea about what devices you want to monitor, what kind of information you need from these devices, and how often you need the information.

## **Taking Over NNM from Someone Else**

If you are responsible for taking over after someone else installed and configured NNM, browse through the chapters in this book in the order in which they are presented. As you read along, rather than configuring NNM, simply access all the configuration points as they are described to determine how NNM is currently configured. Once you understand the current configuration, you can go back through the process and implement your own ideas.



## Understanding the Network Information Needs of Your Organization

To use NNM to its full potential, start with a clear idea of the business needs that NNM must address. You need to understand the groups within your organization and their information needs. You also need to start with a good understanding of your network itself.

### Who Needs What Information?

First identify the groups within your organization who need access to information about the network. For example, upper management, facilities, technical support, and the printers group.

Then, identify what information each group needs, why they need it, and how they will use the information. For example, do they need real-time monitoring of specific devices around the clock or only during specific shifts for network troubleshooting? Do they need to know monthly trends of the volume of network traffic generated by each department for planning purposes?

Use the work sheet provided (page 67) as a starting point and detail the information needs of each group. Then decide which groups could share a map. Have a clear idea about what information everyone needs and how they want to receive that information before you start.

### Ideal Uses for Maps

- At-a-glance notification that something has gone wrong.  
NNM's color-coded status notification alerts your team members immediately when a network or device is in trouble.
- Connectivity at-a-glance.  
A visual untangling of your network, the definitive authority of how everything is connected. The map is kept up-to-date automatically.
- Simultaneous monitoring of multiple sites.

NNM can span multiple sites. In addition to NNM's map and alarm list, NNM's pull-down menus provide troubleshooting tools that allow your team to gather information and troubleshoot problems at the remote site.

- Quick response time.

NNM's maps can be configured with floor plans of your site to speed your team's efforts in investigating hardware failures. If your team has a high turn-over rate, well designed maps can reduce training time.

- Establishing sphere of influence or management regions.

You can have different maps for different areas of responsibility on your network. A map can manage a specific physical portion of your network or partition information about your network. Maps can have specific constraints and characteristics, run different applications, cover different geographic areas, and/or set compound status differently. You can customize maps so that they become a projection of your team member's responsibilities, showing what your team member wants to see about your network without requiring him/her to sifting through unneeded data.

### **Ideal Uses for the Alarm Browser**

- Instant awareness of failures.

NNM's Alarm Browser guarantees the speediest possible response, since your team members know instantly when trouble arises.

- Watching trends.

Configure NNM to post threshold alarms from your mission-critical devices. Your team members detect developing trouble and start resolving it before users encounter network failures.

- Identify the device that is most likely causing the problem.

NNM's event correlation system identifies the source device of certain issues and consolidates all alarms that were generated as a result of the root cause issue under the main alarm message.

- Automate responses.

You can configure NNM to automatically page or email a specific person whenever an alarm is received about a specific device. You can also configure a pop-up window to appear over the NNM map if a certain event occurs or launch any customized action that you wish.

- Facilitate team communication to eliminate duplicate efforts.

Your support team can use NNM's Alarm Browser as a prioritized to-do list. Each team member can select issues to work on and change the alarm's status to *Acknowledged*. Everyone on the team will see immediately that the alarm is being addressed.

- Distribute responsibility.

You can set up custom categories for incoming alarms and configure specific devices to post their alarms to your custom categories. For example, configure the alarms generated by devices in each department to post their alarms to a category identifying the department. Your team members can access the alarm list that applies to them, without needing to sort through alarms for all departments.

## Work Sheet for Planning

Group/user: \_\_\_\_\_

Role/purpose: \_\_\_\_\_

Organization/Location: \_\_\_\_\_

Hours/shift: \_\_\_\_\_

Email address: \_\_\_\_\_

Phone number: \_\_\_\_\_

What do they need to know?

How often do they need this information?

Why do they need to know?

Which devices do they need to monitor?

What type of system will they use to access NNM?

Which MIB value or MIB expression would provide access to this information?

What would be the most beneficial form in which to receive the information?

- Map displayed on a monitor (map configuration)
- Alarm list (event configuration)
- Email (automatic action programmed through Event configuration)
- Pager message (automatic action programmed through Event configuration)
- Web site access
- Reports

## Knowing Your Network

You probably maintain lists of the devices that are currently installed on your network. Use your current lists when first installing NNM in order to verify that everything has been successfully discovered. Your lists will also be useful while deciding how often NNM should monitor each device for you.

### Deciding to Manage or Not to Manage Devices

Because NNM is actively polling your network devices over standard network protocols (ICMP, SNMP, and IPX), a certain amount of network traffic is generated for each managed device. You need to determine the appropriate polling frequency in your business environment. Identify the mission-critical devices on your network, typically your routers, switches, hubs, and key servers. You will want to configure NNM to monitor these devices most frequently. NNM allows you to configure groups of devices by using wildcards or a number of other identifiers. Therefore, if you have a clear idea of where you are heading, the configuration process will be much easier and quicker.

For planning purposes, consider grouping your network devices into three network management categories:

- Mission-critical devices.

You will monitor these devices aggressively, perhaps configuring NNM to poll them two or three times per minute to verify that continuous service is available. You may want to use NNM's event configuration to maximize the use of information available through these devices' SNMP agents and MIB files. Also consider collecting performance data for later trend analysis. Balance your desire for data and polling against the overhead you are imposing on devices.

- Important, but not critical.

You will configure NNM to monitor these devices less frequently: perhaps your printers, certain application servers, firewalls, http proxies, etc. Depending on the situation at your site:

- Every few minutes?
- Every few hours?

- Every few days?
- Devices that do not require active monitoring.  
Not important enough to warrant generating network traffic (for example individual PCs). Therefore, you can disable NNM's polling of these devices by setting them to *unmanaged* or you can configure polling to run very infrequently.  
Although you configure certain devices or groups of devices to be *unmanaged* by NNM, the device's SNMP agent can be configured to forward traps to the management station if anything goes wrong. These traps will appear as alarms in NNM's Alarm Browser once you configure them to do so by using NNM's event configuration feature.  
NNM can be configured either to include the unimportant devices on your maps or to exclude them from the map.

### Work Sheet for Devices

Identify which devices on your network fit into the following groups. Your list will come in handy when you are troubleshooting NNM's initial discovery, when you are customizing NNM's polling configuration, and when you are creating customized maps for your team.

- Mission-critical devices.  
Must be frequently monitored on a proactive basis to ensure continuous service.
  
- Important (but not critical) devices.  
These devices can be monitored less frequently than mission-critical devices. For example: you may want NNM to monitor some devices every 30 minutes, or every 12 hours, or every 7 days. Decide upon meaningful monitoring frequencies for different groups of your network devices

- Devices that do not require network management.  
These devices do not require active monitoring (their SNMP agents can be configured to forward traps to the management station if anything goes wrong).

---

**TIP**

---

Network devices that did not show up on anyone's worksheet from the previous section in this chapter probably do not need to be actively managed by NNM.

## Starting with a Well-Configured Network

As explained in Chapter 2, “How Does NNM Work?,” on page 31, NNM communicates with the devices on your network through standard network protocols. For NNM to gather accurate information, your network needs to be in a well-configured state so that each device can be successfully accessed.

The following aspects of a well-configured network do not need to be in perfect shape on your network before you install NNM. However, the closer you are to having a well-configured network, the quicker and easier it will be to configure NNM. Relevant issues include:

- Consistent IP addressing scheme
- Consistent subnet masks configured
- DHCP address ranges identified
- Consistent name resolution scheme (such as DNS or NIS)
- SNMP agents configured on each device:
  - Know the agent’s GET- and SET-community names
  - Update the trap-forwarding (trap destination) list to include the NNM management station
  - Know which MIB is being used and obtain a copy of it to load into the NNM management station’s MIB database later on

### Consistent IP Addressing Scheme

A list of your *network-level* IP addresses would be handy to have during the initial discovery process so that you can quickly verify that NNM has successfully discovered everything you need. This list could also be used to create a seed file to speed the discovery process.

Remember that all devices that you wish to manage must have a valid IP address. For hubs and other network devices that normally do not have IP addresses and enabled SNMP agents, it is often possible to manually establish them. Determine if the device can support SNMP, then physically visit each device or access it through a terminal and take the necessary steps to assign an IP address and enable the SNMP agent. Contact the device’s vendor if you need more information.



## Subnet Masks Consistently Configured

If the subnet masks on the management station or configured on your network devices are incorrect, discovery may be different from what you expected. If the subnet mask is too restrictive, you won't discover your whole management domain. If the subnet mask is not restrictive enough, you discover more than you bargained for.

Non-contiguous subnet masks (for example, 255.255.0.255) are not supported by NNM. Also, in hierarchical subnets, two subnets having the same IP subnet address but different subnet masks are not supported. See the white paper about subnet masks ("White Papers" on page 57) for more information.

---

### TIP

NNM does not detect proxy ARP (an alternative to defining extensive routing tables). If NNM is running in an environment using proxy ARP, alarms about misconfigured subnet masks are generated. In this case, these alarms can be ignored.

---

## DHCP Address Ranges Identified

HP strongly suggests that you assign a range of IP addresses to be randomly assigned to mobile computers when they access your network. NNM can handle this range of addresses in a way that does not cause problems as users hook up to and disconnect from the network.

## Consistent Name Resolution Scheme

NNM depends upon your network name resolution scheme being in good working order (DNS, NIS, NIS+, NetBios, local hosts files). NNM cannot conduct successful discovery if it is receiving inaccurate information. NNM will exhibit performance problems and erratic behavior when your name resolution scheme is not working. If you have name resolution in place prior to NNM's initial discovery process, hostnames are used for map symbol labels, rather than IP addresses. Hostnames generally make the NNM maps and alarm messages more meaningful to your team.

If you are using DNS, it is recommended that the management station be a caching or secondary name server. Ensure that the management station secondary name server is not serving any other clients.

---

**NOTE**

Point your web browser to <http://www.docs.hp.com> and search for a document titled *Installing and Administering Internet Services* (part number B2355-90147). This document contains information about configuring a caching-only name server.

---

### Quick Check of Your Name Resolution Scheme

To test the health of the name resolution implementation before installing NNM, try the following procedure.

1. Find the `gethost.exe` tool (provided on the NNM installation CD) and copy it to the hard drive of the machine where you will install NNM.

*Windows:* `CD-ROM drive:\support\gethost.exe`

*UNIX:* Mount the CD drive,  
`/cdrom/OVDEPOT/OVNNMgr/OVNNM-RUN/opt/OV/support/gethost`

2. Choose an IP address that is not in use on your network.
3. Navigate to the directory where you copied `gethost`. At the command prompt type either:

**`gethost -a -i bogusIPAddress`**

**`gethost -i realHostName`**

4. If you receive an immediate answer, your name resolution scheme is in good shape. If your system hangs while trying to formulate an answer, clean up your network name resolution before installing NNM.

---

**NOTE**

To test the health of the name resolution implementation after installing NNM, see “Opening NNM takes a long time” on page 116.

---

## SNMP Agents and Their MIBs Configured

The SNMP agent on each network device responds to GET and SET requests from NNM and continuously monitors the device upon which it resides. The agent generates and sends SNMP traps to NNM whenever an error condition arises that is defined in the agent's MIB file.

Configure each SNMP agent's community names and trap destination list specifically for your particular network. Obtain a copy of the agent's MIB file so that you can configure NNM to take full advantage of all the management capabilities provided by the agent.

If an SNMP agent is not currently installed on a network device that you need to monitor, contact the vendor of the device.

### GET- and SET-Community Naming Scheme

Community names are a lightweight security technique employed by SNMP agent software. If the network management software wants to communicate with the SNMP agent software, both must use the same community name during GET and SET requests and talk over the same port. By default, community names are often set to PUBLIC and the port is set to 161. NNM assumes that these defaults are in use unless you inform it otherwise. Many network administrators change the GET-community name so that hackers cannot obtain sensitive information about their devices. It is even more common for network administrators to change the SET-community name, since knowing this community name enables others to make system configuration changes to the device itself, either inadvertently or maliciously.

In order to work, NNM depends upon successful communication with the SNMP agents installed on your network devices. If the community names have been changed from PUBLIC or the port changed from 161, you need to inform NNM of the new ones. (NNM cannot obtain these automatically because that would defeat the security feature.) NNM tests community names in the file `netmon.cmstr`. See "GET- and SET-Community Name and SNMP Port Issues" on page 128.

### Trap Destination List Updated to Include the NNM Management Station

The SNMP agent software continuously monitors its device for error conditions defined in the installed MIB. When an error condition is detected, a trap message is automatically forwarded to all devices listed in the agent's trap-forwarding list. When configuring the SNMP agent on

each network device, configure the agent's trap forwarding list (or trap destination list) to include the NNM management station's host name or IP address. NNM will then receive traps from the device and keep you informed. Refer to the agent's documentation for information about how to do this. See "Ensuring that NNM Receives Traps from Your Network Devices" on page 311.

---

**TIP**

*HP-UX systems only:* For hosts with HP OpenView SNMP agents, NNM automatically updates the trap forwarding list when the device is discovered, provided that the correct SNMP SET-community name is configured within NNM. See "GET- and SET-Community Name and SNMP Port Issues" on page 128.

---

### **Obtain a Copy of the SNMP Agent's MIB File**

Often vendors provide custom MIB files that define specific objects for monitoring the particular device. Be sure to request a copy of the custom MIB so that you can install it into NNM's MIB database later on.

---

**NOTE**

List of all of the MIBs in use on your network devices. These same MIBs will need to be installed on the NNM management station. See "Loading MIBs in the MIB Database" on page 395. After you install the MIB you can configure NNM to handle each trap as you wish. See "Event Configuration Overview" on page 404.

---

To obtain the most recent version of a MIB file, check NNM's installation CD, contact the vendor who wrote the MIB file, or search the website <ftp://ftp.isi.edu/mib/> for quick access to the MIB file. Make sure the MIB file that you load onto the management station's MIB database is the same version being used by the SNMP agent on your network device.

### **DMI Service Providers and Their MIFs Configured**

If your management station uses a Windows operating system, NNM **automatically subscribes** to receive DMI events from all managed systems running DMI version 2.0.

Keep track of any vendor-specific MIF files being used on these devices. Your list will come in handy later in NNM's configuration process. If your DMI service provider software is using a vendor-specific MIF instead of or in addition to the DMTF's standard MIF, during NNM's configuration process you must configure NNM to understand the custom MIF events.

For information about configuring DMI MIFs, see "DMI Events (Indications)" on page 313, "DMI Browser" on page 399, and "Loading DMI-to-SNMP Event Mappings" on page 401.



---

---

# 5

## **Initial Network Discovery: Options and Troubleshooting**

The first step toward configuring Network Node Manager (NNM) is to install the software and allow initial discovery to run.

NNM automatically discovers each device on your network and its connective relationship to other devices. NNM's object and topology databases store the discovered information and NNM uses this information to draw the default map and set up the event tracking system. After initial discovery has run to completion, you use the map to identify and fix problems on your network.

Often, network administrators with routed environments simply start NNM and allow it to run initial discovery with the default settings. By default, your management domain consists of the network indicated by your *management station's IP address and subnet mask*. NNM discovers everything to the edges of that network and **manages** (monitors) the devices discovered up to the number in your license agreement limit.

If you need to manage multiple networks, you can *add* them to the management domain by:

- Manually selecting symbols on the map and setting them to *managed* (see “Interactively Expand/Limit Your Management Domain” on page 100).
- Automatically setting devices on multiple networks to *managed* by creating and using a seed file, using the `loadhosts` program, and/or increasing the IPX-Hop Count (see “Automatically Expand Your Management Domain” on page 102).

In very large enterprises and bridged environments, the initial discovery process may detect such a large number of devices that managing them all would exceed your organization's and your management station's resources. In this case, you may want to *limit* the number of devices included in the initial discovery process. You can limit initial discovery so that it only includes your organization's mission-critical devices by:

- Manually selecting unimportant symbols on the map and setting them to *unmanaged* (see “Interactively Expand/Limit Your Management Domain” on page 100).
- Creating and using a `netmon.noDiscover` file that specifies which devices to *exclude* from the discovery process, using the `loadhost` program to discover only the devices on your list, creating and using a *discovery filter* that controls which devices or types of devices to



*include* in the discovery process, using the `oid_to_type` file to unmanage groups of devices, and/or decreasing the IPX Hop Count (see “Automatically Limit Your Management Domain” on page 107).

If your network is quite large, but you don’t want to eliminate devices from the management domain, you can visually break the map into smaller units or create multiple maps that draw information from the same databases. You can use the hide feature or map filter to show different aspects of your network on each map. See Chapter 8, “Map Customization,” on page 197.

If you purchased the NNM Advanced Edition, it is also possible to use multiple management stations in a hierarchical arrangement that divides up the management responsibility yet shares critical information. See *A Guide to Scalability and Distribution* for more information.

You start benefiting from NNM immediately. This initial map discovery process may reveal configuration problems on your network, such as SNMP agents that are not communicating properly, incorrect subnet masks or DNS settings, mismatched community names, and unexpected connections to network segments that you do not own. The map is a visual representation of the communications channels established between NNM and the devices in your network. The accuracy of these links determines whether or not NNM can provide valuable information to help you manage your network. Be sure to follow the directions for verifying the accuracy of your maps (see page 92) and to fix any problems before continuing with further customizations to NNM (page 114).

---

**TIP**

Your company may have purchased multiple HP OpenView products. If you are using other HP OpenView applications in addition to NNM, you may notice that the menu choices and options provided within the software vary from what is presented in this book. This is because the other HP OpenView applications integrate closely with NNM to take advantage of the features that NNM already provides. Other HP OpenView programs build upon the foundation provided by NNM.

---

## Install Network Node Manager (if you have not already done so)

See the *Quick Start Installation Guide* included with the NNM Installation CD for instructions about how to install Network Node Manager. Make sure that your management station meets the minimum requirements before you start.

---

### NOTE

In this manual, C and \$LANG in directory listings are variables for the language currently in use. C contains the English version.

---

## Installing IPX Transport Software (Windows only)

*For management stations running a Windows operating system:*

IPX discovery is an optional component of Windows operating systems. If you want to discover and monitor IPX nodes, be sure to install the NWLink IPX/SPX Compatible Transport network software stack before or during the NNM installation process. These drivers are included with the Windows operating system installation software, but are not installed by default. See the *Quick Start Installation Guide* included with NNM's installation CD for more information.

## Establishing Universal Pathnames

This book uses directory pathnames that are independent of the underlying file structure of the operating system you are using (see your data sheet to learn specifically which operating systems and versions this release supports). For each Network Node Manager directory, a single pathname is used rather than multiple pathnames that vary by system. The instructions you use to set up these universal pathnames vary depending on the operating system you are running.

## Establishing Universal Pathnames on a UNIX Operating System

*For management stations running a UNIX operating system:*

The following text shows an example of a universal pathname: a pathname for some common NNM log files is stated as `$OV_LOG` rather than `/var/opt/OV/share/log`. For a full list of these pathnames, see the `$OV_BIN/ov.envvars.sh` file.

Take the following steps to set up each user's environment. Setting the universal pathnames and paths as shown is temporary; you will need to do this every time you start a terminal window. Or you can permanently set each user's path by modifying the path in the appropriate files shown below:

If running `sh` or `ksh`: Modify the path in the `.profile` file.

If running `csh`: Modify the path in the `.login` file.

If running `VUE`: Modify the path in the `.vueprofile` file.

1. If you want to use the universal pathnames that have been incorporated into NNM, you need to *source* the file into your current environment. Use one of the following commands, depending on which shell you are running. If you are not sure which shell you are running, execute:

```
echo $SHELL
```

---

**NOTE**

---

Be sure to include a space between the `.` and the `/` in the `sh` or `ksh` commands.

**If running `sh` or `ksh`:**

```
On UNIX:      . /opt/OV/bin/ov.envvars.sh
```

**If running `csh`:**

```
On UNIX:      source /opt/OV/bin/ov.envvars.csh
```

See the `ov.envvars` manpage for additional information about this script and information about the universal pathname equivalents.

2. Set the path to include `$OV_BIN` and `$OV_MAN`. Execute the following commands:

**If running `sh` or `ksh`:**

## Install Network Node Manager (if you have not already done so)

```
PATH=$PATH:$OV_BIN
MANPATH=${MANPATH:-/usr/man:/usr/share/man}:$OV_MAN
export PATH
export MANPATH
```

### If running csh:

```
set path=($path $OV_BIN)
if (${?MANPATH} == 0)setenv MANPATH "/usr/man:/usr
/share/man"
setenv MANPATH "${MANPATH}:$OV_MAN"
```

---

### NOTE

All NNM documentation referring specifically to a UNIX operating system assumes that you have activated the universal-pathname environment variables.

---

For example, a written procedure might include the following step:

```
cd $OV_BACKGROUNDS
```

This command takes you to the local directory containing submap backgrounds, and it works regardless of which version of the UNIX file system is involved. Note that if you need to know the actual pathname, you can always enter a command like the following:

```
echo $APP_DEFS
```

The response to this command will be the actual local pathname:

- *HP-UX*: /usr/lib/X11/app-defaults
- *Solaris*: /usr/openwin/lib/app-defaults
- *Linux*: /usr/openwin/lib/app-defaults

## Establishing Universal Pathnames on a Windows Operating System

*For management stations running a Windows operating system:*

The following text shows an example of a universal pathname: a pathname for some common NNM log files is stated as %OV\_LOG% rather than install\_dir\log. For a full list of these pathnames, see the %OV\_BIN%\ov.envvars.bat file.

If you want to use the universal pathnames that have been incorporated into NNM for the Windows operating system, you need to run the following commands:

1. `cd install_dir\bin`
2. `ov.envvars.bat`

## Discover the Network... Let NNM Do It

Let NNM do the work discovering your network for you. NNM sets up several databases, creates the default map, and establishes communication links for tracking events on your network.

### Overview

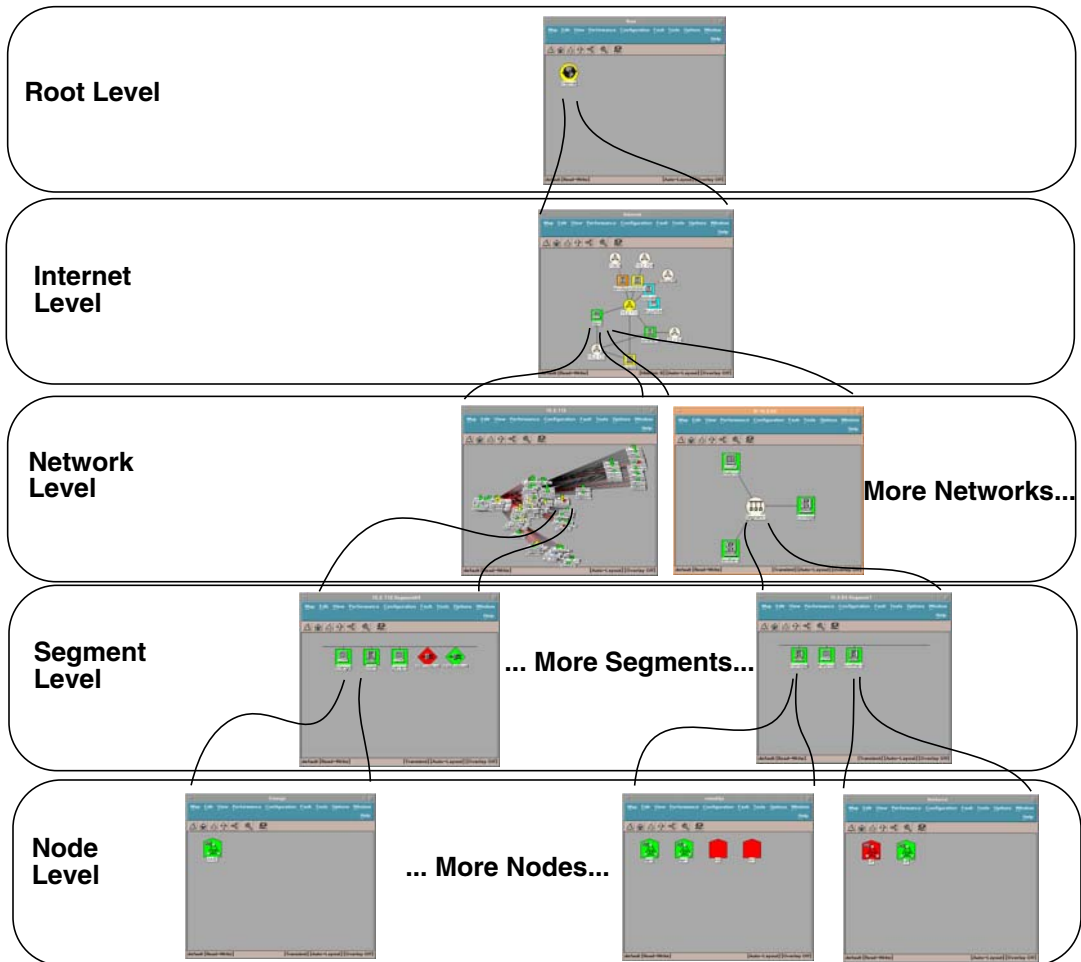
First NNM's services (background processes) are started. Expect intense polling traffic at first as the network monitoring service, `netmon`, works as quickly as possible to discover each device on your network. An ICMP ping is issued for each device identified in any found ARP table and an `snmpwalk` command is issued to gather information about each discovered device. If a serial WAN link is discovered, NNM checks the router's routing table for clues about how to continue with discovery. The frequency of polling for new nodes automatically decreases as fewer new objects are discovered per polling cycle.

NNM discovers the following objects and places them on the default map:

- **Internet-level submap:** IP networks, gateways, routers, and multi-homed workstations
- **Network-level submaps:** bus, star, and ring segments; gateways, routers, switches, hubs, and bridges; *if using a Windows operating system add:* IPX networks, gateways, and routers
- **Segment-level submaps:** hosts, gateways, routers, switches, hubs, and bridges; *if using a Windows operating system add:* IPX networks, gateways, and routers

- **Node-level submaps:** network interface cards

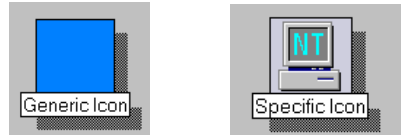
**Figure 5-1** A Typical Hierarchy of Persistent Submaps



NNM uses the information about each device gathered by `netmon` to determine the most accurate and descriptive symbol to use on the map. If NNM cannot match the `sysObjectID` to a specific symbol type, the node will be represented as a generic symbol on the map (a shape without a specific descriptive icon inside of the shape). For example, some IP nodes

that are not computers (connectors such as bridges and repeaters) may initially appear on the map as hosts. See “Troubleshooting Discovery” on page 114 for more information.

**Figure 5-2**      **Generic Computer Icon versus Specific Descriptive Icon**



---

**NOTE**

NNM determines which symbol to use for each device by referencing information contained in the following configuration file and directories:

- *Windows* directory:  
`install_dir\conf\oid_to_sym_reg`
- *Windows* file:  
`install_dir\conf\oid_to_type`
- *UNIX* directory  
`$OV_CONF/oid_to_sym_reg`
- *UNIX* file  
`$OV_CONF/oid_to_type`

The directories mentioned above show the top level of a directory structure that organizes information by vendor. Although the entries in these files are already extensive, you may want to make additions or changes to these files. For example, you may want to make changes if you see a generic icon on the map and wish to assign a specific icon. Another example is making additional entries if you want to create your own new icons. See the instructions at the beginning of each of these ASCII files. See also the *oid\_to\_type* and *oid\_to\_sym* reference pages in NNM’s online help (or the UNIX manpages) and Appendix C, “Changing All the Symbols for a Particular Device,” on page 623 and Appendix D, “Changing an Object’s Vendor and SNMP Agent,” on page 637 for more information.



## Start the NNM Services (background processes)

NNM starts collecting the information needed to draw the default map as soon as you start the NNM services. The services continually monitor the network and track activity. You need to start them the first time you use NNM; thereafter, they run continually whether or not you have the NNM user interface open. To learn more about these services, see Appendix A, “NNM Services and Files,” on page 523.

To start the services (background processes):

- *Windows:*  
Select Start:Programs:HP OpenView:  
Network Node Manager Admin->NNM Services - Start.
- *UNIX:*  
As root, at the command prompt, type **ovstart**  
  
The `$OV_BIN` directory must be in your `PATH`. If not, you must type **`$OV_BIN/ovstart`** to start NNM.  
  
See the `ov.envvar` manpage for a complete list of the universal pathnames and their equivalents, and information about a script for setting up environmental variables that will create the pathnames on your system.

## Make Sure NNM's Services Are Successfully Running

To generate a status list of the services that make NNM work:

- *Windows:*  
Select Start:Programs:HP OpenView:Network Node Manager  
Admin->NNM Services - Status
- *UNIX:* At the command line prompt, type **`ovstatus -c`**.

---

**TIP**

---

Type **`ovstatus -c`** for concise, one-line-per-process information.

If any of the services are not successfully running, see Appendix A, “NNM Services and Files,” on page 523 for information about troubleshooting services before continuing.

## Open NNM

Opening NNM takes a few minutes as the maps are synchronized with the databases. To start NNM's user interface (foreground processes):

- *Windows:*  
Select `Start:Programs:HP OpenView:Network Node Manager`. NNM automatically starts all installed and registered applications.
- *UNIX:* At the command prompt, type `ovw`. NNM automatically starts all installed and registered applications.

## Configure an Inventory Report

Use the web Reporting interface to configure an inventory report. An inventory report lists the network nodes that are contained in the topology database, organized by subnet. You can use this report to help verify the accuracy of the initial discovery of your network. Refer to the online help for more information on configuring an inventory report.

## Let NNM Run Over Night to Populate the Maps

The initial discovery process takes time. We recommend that you let NNM run over night to gather the majority of network information available. As NNM maps your network, it constantly rearranges icons. Therefore, it is best to wait until initial discovery has finished before exploring the default map.

NNM gathers information to the limits of the network indicated by the management station's IP address and subnet mask. NNM then checks the ARP table on each discovered device and expands the map by adding devices listed in each ARP table to the map. NNM also detects serial WAN links (for which ARP caches have no entries) and then checks the resident routing table instead. Devices listed in the routing table are then discovered and the process continues. NNM follows this process each time communication is established with a device, so your maps will continue to grow over time. (See Chapter 2, "How Does NNM Work?," on page 31 for a more detailed explanation about how NNM discovers network information.)

---

**NOTE**

While NNM discovers and maps your network information, explore the *Get Acquainted with HP OpenView Network Node Manager: Training for NNM Operators* CD provided with NNM so that you will understand what you are seeing after the discovery process is complete.

---

After NNM completes initial discovery, verify the accuracy of the map (see page 92), making sure that any devices that you have identified as mission-critical in Chapter 4, “Planning Your NNM Configuration,” on page 61 appear on the map. Then fix any problems by:

- Interactively expanding or limiting your management domain (see page 100).
- Automatically expanding your management domain (see page 102).
- Automatically limiting your management domain (see page 107).
- Troubleshooting the initial map to ensure your network is configured properly (see page 114).
- Delete NNM’s object and topology databases and restart discovery, if necessary (see page 144).

When you are satisfied with the accuracy and completeness of the initial map, go to Chapter 6, “Preserve Your Sanity: Backup and Polling Configuration,” on page 147 to learn how to back up your work and to customize the polling configurations.

## Verify the Accuracy of the Initial Discovery

Now that NNM has had time to map all the network connections discovered, take the time to make sure you recognize what was found. Refer to the checklists that you developed in Chapter 4, “Planning Your NNM Configuration,” on page 61 to verify that all mission-critical devices are present.

Usually there are a few surprises on your map. NNM allows you to easily identify such things as connectivity problems and security risks. Take the time to fix any problems that you discover (see “Troubleshooting Discovery” on page 114).

This section includes information about NNM’s tools that make verifying the information on your maps easier:

- Inventory report  
View or print the inventory report.
- Object properties information  
View a detailed description of the underlying object properties for the selected internet, network, segment, node, or interface symbol.
- Find object feature  
Locate map elements quickly using the `Edit:Find` menu item.

---

### TIP

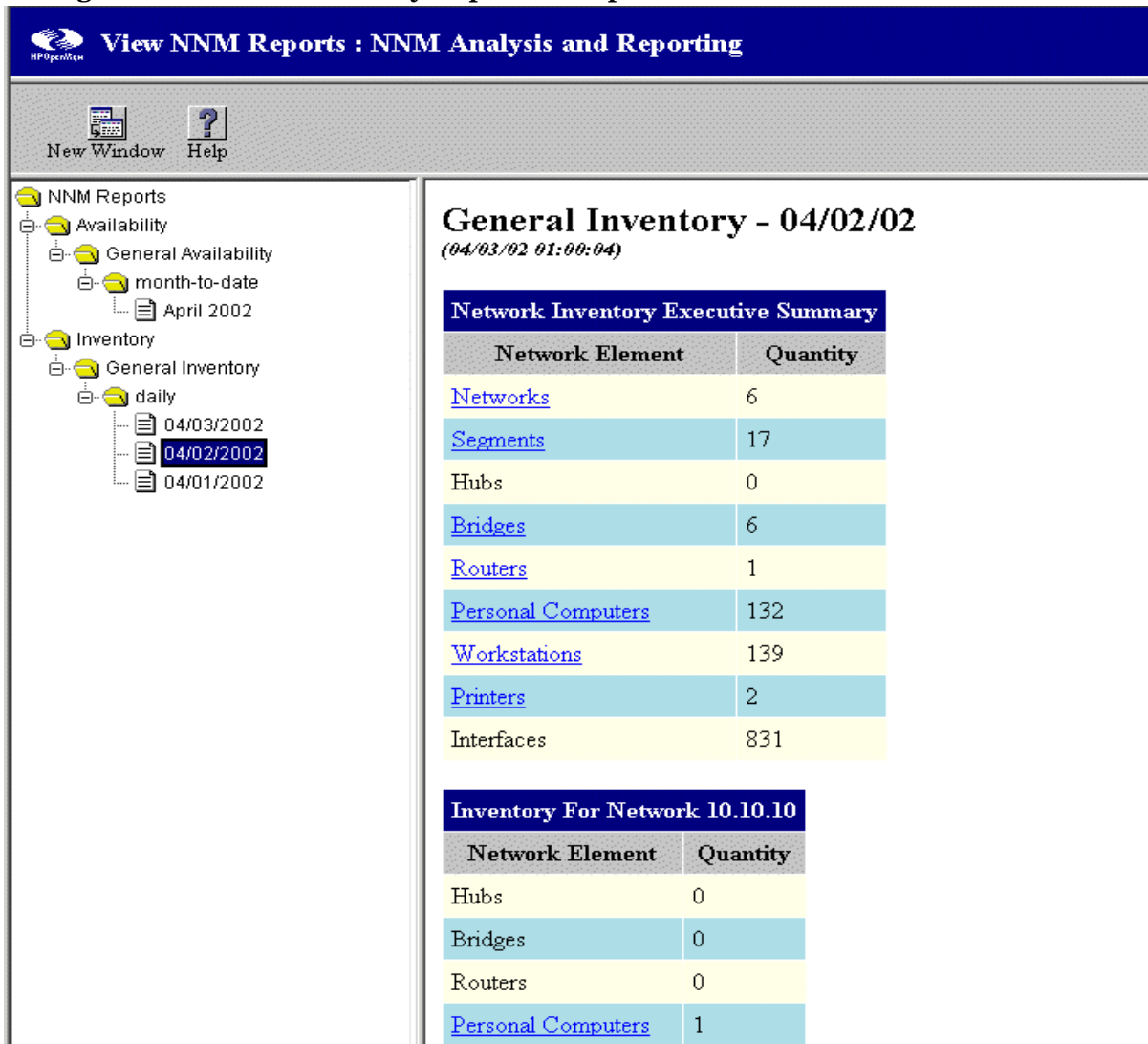
You can generate an ASCII file that lists objects discovered by NNM. Refer to the *ovtopodump* reference page in NNM’s online help (or the UNIX manpage) for information. Then use your favorite tools to search or sort the list to confirm that your devices have been discovered.

---

## Printing the Inventory Report

Use the inventory report to see a detailed list of what was discovered on your network. The inventory report lists the network nodes that are contained in the topology database, organized by subnet. An example of an inventory report follows.

**Figure 5-3** Inventory Report Example



The screenshot displays the HP OpenView NNM Reports interface. The title bar reads "View NNM Reports : NNM Analysis and Reporting". Below the title bar are icons for "New Window" and "Help". The left pane shows a tree view of reports under "NNM Reports", with "General Inventory" selected and "04/02/2002" highlighted. The main pane displays the "General Inventory - 04/02/02" report, dated "04/03/02 01:00:04".

**Network Inventory Executive Summary**

Network Element	Quantity
<a href="#">Networks</a>	6
<a href="#">Segments</a>	17
Hubs	0
<a href="#">Bridges</a>	6
<a href="#">Routers</a>	1
<a href="#">Personal Computers</a>	132
<a href="#">Workstations</a>	139
<a href="#">Printers</a>	2
Interfaces	831

**Inventory For Network 10.10.10**

Network Element	Quantity
Hubs	0
Bridges	0
Routers	0
<a href="#">Personal Computers</a>	1

From the Network Inventory Executive Summary, you can click on entries in the Network Element column to see more details.

See the NNM online help for more information on the web Reporting interface.

## Viewing the Properties of Your Network Configuration

You can gain quick access to the IP Map service's information that describes your network and system configurations. The information in the dialog boxes for object properties can help you determine if everything is present and accounted for on the initial map.

### Viewing Object Descriptions

You can view the object properties for any network element that NNM manages, such as an internet, network, segment, node, or interface. The object database maintains this information. Object properties are always as current as NNM's last polling cycle.

**Description of an Internet Object** Simply select the Internet symbol on the root submap and right-click with the mouse. Select *Object Properties* and select *IP Map*. See NNM's online help for more information and specific directions. NNM provides the following information:

- number of networks on the internet
- number of segments on the internet
- number of nodes contained within the internet
- number of interfaces contained within the internet
- number of gateways/routers on the internet
- any status or error messages may appear in the *Messages* field

**Description of a Network Object** Simply select any network symbol on the map and right-click with the mouse. Select *Object Properties* and select *IP Map*. See NNM's online help for more information and specific directions. NNM provides the following information:

- the network name as it appears on the map
- network address

- subnet mask associated with the network number
- status of the network
- number of segments on the network
- number of nodes contained within the network
- any status or error messages may appear in the Messages field

**Description of a Segment Object** Simply select any segment symbol on the map and right-click with the mouse. Select `Object Properties` and select `IP Map`. See NNM's online help for more information and specific directions. NNM provides the following information:

- status of the segment
- number of nodes on the segment
- any status or error messages may appear in the Messages field

**Description of a Node Object** Simply select any node symbol on the map and right-click with the mouse. Select `Object Properties` and select `IP Map`. See NNM's online help for more information and specific directions. NNM provides the following information:

- hostname that was assigned when the node was initially discovered
- status of the node
- information about and status of each interface installed on the node
- system description as returned by the SNMP agent (\*)
- system location and system contact as returned by the SNMP agent (\*)
- system object ID of the node (\*)
- any status or error messages may appear in the Messages field

\* The node must support SNMP to return this information.

**Description of an Interface Object** Simply select any interface symbol on the map and right-click with the mouse. Select `Object Properties` and select `IP Map`. See NNM's online help for more information and specific directions. NNM provides the following information:

- address of the interface

## Verify the Accuracy of the Initial Discovery

- subnet mask (IP only)
- link-level address (physical address)
- interface type
- status
- any status or error messages may appear in the Messages field

### Finding Specific Devices on the Map

The `Edit:Find` menu item enables you to search for objects that contain a common characteristic. You can:

- Find objects in either the open map or an open snapshot based on the value of a specific attribute.
- Create a selection list of objects with particular characteristics, such as all routers.
- Open the submap of each found object.

You can find objects based upon these object and symbol characteristics (see page 191 for more information):

- Selection Name

You can find objects based on a selection name or part of a selection name.

- Attribute

You can search for objects based on the values of chosen attributes, such as `isRouter`.

- Comments

You can search for all objects that have a specific comment or portion of a comment. You may establish conventions for embedded keywords and use these in the `Find By Comment` operation. In this manner, you can find objects based on a key that is not a supported attribute.

- Symbol Status

You can find objects that are represented by a symbol with the given status. Each found symbol is listed separately.

- Symbol Type



You can find all objects that have a symbol of a given type (class and subclass).

- Label

You can search on the label string (comments added by you).

### Entering Strings in Find Operations

When you select `Edit:Find`, input in the form of a string value is sometimes required. For several operations, you have a choice between two types of searches: exact match and pattern matching.

**Exact Match** Exact match finds only those objects whose attribute value exactly matches the string entered. To switch to the exact match search, click on the `Exact Match` radio button. You must enter the complete string in this mode.

**Pattern Matching** When pattern matching is used, you may enter a simple substring or a regular expression. The following section, “Regular Expressions,” discusses some of the special characters you can use in regular expressions.

A valid regular expression may consist completely of printable characters. However, in a regular expression, certain symbols have special meaning; see the following table. (*HP-UX only*: more regular expressions are available, see the *regex* manpage.)

**Table 5-1 Regular Expressions for Pattern Matching**

<b>Special Character</b>	<b>Meaning</b>	<b>Examples</b>
. (period)	Matches any single printable or nonprintable character except <newline> (unless it is used inside brackets).	node.in matches: node3interface nodeXinterface

**Table 5-1** Regular Expressions for Pattern Matching (Continued)

<b>Special Character</b>	<b>Meaning</b>	<b>Examples</b>
* (asterisk)	Means zero or more occurrences of the preceding character.	node* matches: nod node nodeTwo nodee
^ (Caret)	If the caret is the first symbol, then the following string must be the first part of any matching string.	^Box matches the string: Box elder but does not match the string: Big Box
\$(dollar)	If dollar is the last symbol, then the preceding string must be the last part of any matching string.	long\$ matches the string: belong but does not match the string: longer

### Results of the Find Operation

NNM reports all found objects to you in the dialog box. The Message area reports the results of the search. It will display:

- the number of objects found
- errors
- further instructions

Symbols of found objects are listed by selection name in the Located and Highlighted area. If a found object is represented by multiple symbols, each symbol is displayed on a separate line.

**Table 5-2 Results of the Find Operation**

<b>Parameter</b>	<b>Meaning</b>
Selection Name	The selection name of the object.
Submap Name	The name of the submap that displays a symbol of a found object.
Symbol Label	The symbol label as it appears on the submap.

The following three example entries display one found object and three symbols that represent the object in three different submaps:

```
NodeBob:lanCard:IP address [connection to segment]
```

```
NodeBob [connection to segment]
```

```
NodeBob [hostname]
```

The symbols of found objects are highlighted on submaps.

## Interactively Expand/Limit Your Management Domain

The NNM default map includes any devices that have already been discovered. You control which of these devices are **managed** (monitored by NNM's services) and which of these devices are **unmanaged** (placed on the map as a light-tan icon, but not monitored by NNM services). NNM does not actively monitor unmanaged devices through polling. However, you can still configure each unmanaged device's SNMP agent or DMI service provider to forward traps to the NNM management station's event tracking system (see "Ensuring that NNM Receives Traps from Your Network Devices" on page 311).

Normally, you customize the size of your management domain by interactively *managing* devices from other networks to expand it and/or *unmanaging* non-critical devices to limit network management traffic and the size of your map.

---

### TIP

If this approach seems too time consuming based upon the size of your management domain, see page 102 and page 107.

---

## Adding Networks to Your Management Domain

To add a network to your management domain, find the IP address and hostname of at least one high-traffic device within the network that you wish to add. Gateways, routers, or multi-homed servers make the best choice. Then:

- On the Internet-level submap, select the `Edit: Add Objects` menu item.
- Select the *Connector* Symbol Class icon, then drag the *Gateway* Symbol Subclass icon onto your Internet-level submap. Choose this gateway connector regardless of the type of device you are using to start the discovery. The `Add Object Palette` appears.
- Double-click on `IP Map` to open the `Set Attributes` dialog box.

- Type in the IP address and hostname of an SNMP enabled device within the network that you wish to add to your management domain, then click on the Verify button.

Once NNM runs the configuration check, NNM will correct the symbol choice and (if necessary) placement for you. The device is configured to be managed by NNM.

## Managing an Unmanaged Network Device

To change the status of a network device from unmanaged to managed (see NNM's online help for more information):

1. From any submap, use NNM's `Edit:Find` menu item to locate the device's symbol on the map.
2. Select the device's symbol and select `Edit:Manage Objects`.
3. Select a router that is connected to the new network symbol and use the `Fault:Network Connectivity->Poll Node` menu item.

NNM conducts discovery on that device's assigned network or networks and the nodes within any neighboring networks. NNM updates the map, databases, and event monitoring to include these devices.

## Unmanaging a Managed Network Device

If you *unmanage* a network symbol or segment symbol during active discovery, a few node symbols on that network or segment may not be reset to unmanaged. To avoid this problem while *unmanaging* the network symbol or segment symbol, do one of the following:

- Wait until NNM finishes active discovery before unmanaging an object.
- Turn *new-node discovery* polling off. From any submap, select the `Options:Network Polling Configuration`. Disable the `New Node Discovery` option. Don't forget to turn polling back on again when you are finished modifying the map.

## Automatically Expand Your Management Domain

Normally, `netmon` generates the discovery process outward from the management station and includes only the management station's IP network in the management domain. If your NNM maps are too limited and do not show *all* the networks for which you are responsible, and adding the missing networks one by one (page 100) would be too time-consuming, you have several choices to consider for customizing the discovery process:

- Seed File (page 103)

If your management domain contains multiple networks (especially networks that are not adjacent to each other), consider using a seed file. A **seed file** instructs `netmon` to generate the map starting from multiple IP addresses instead of or in addition to the management station's IP address. This saves you time by eliminating the manual process of expanding the management domain.

- `loadhosts` Program (page 105)

If your management domain contains few SNMP devices, you can speed up `netmon`'s discovery process by creating an ASCII file listing devices that need to be added to the topology database and map. The `loadhosts` program reads this file and adds the devices listed in this file directly to your topology database, bypassing the `netmon` discovery process. Upon being added to the topology database, `netmon` begins polling the devices as newly discovered devices and expands your management domain from these new nodes.

---

### TIP

If you already keep a local hosts file that lists important SNMP-enabled nodes on your network, you can use that file.

---

- IPX Hop Count (Windows only, page 106)

If you installed the Windows drivers for the IPX stack, during discovery gateways and servers acting as routers return a **hop count** (how many hops are allowed through gateways and routers to

get to the network) for each network of which the router is aware. You can expand the extent of discovery by setting the maximum hop count to a higher number.

---

**TIP**

Remember that each additional managed network device increases the network management traffic being generated. Refer to the critical-device list that you established in Chapter 4, “Planning Your NNM Configuration,” on page 61 for guidance about keeping the management polling traffic to a reasonable minimum. Unmanage devices that are not mission-critical.

---

## Create a Seed File Specifying Multiple IP Networks to Manage

A seed file is an ASCII file containing a list of network-level IP addresses or host names (not IPX addresses). The contents of the seed file define your map’s initial management region. Gateways make the best seeds because they maintain the largest ARP tables. If you add the major gateways, `netmon` will discover the rest of the nodes on the network, so you do not have to include all nodes in the seed file. All networks attached to the devices listed in your seed file will be discovered and set to *managed*. Networks implied by each node’s IP address are discovered and added automatically to the map.

1. Create an ASCII file. The name and location of this ASCII seed file is up to you. In this file, list any hostnames or IP addresses that NNM should use to begin discovery. Each hostname or IP address must be on its own line, and each listed device must support SNMP. Specifically, `netmon` must be able to determine the subnet mask for each interface in a specified node. Otherwise, the seed file entry will be ignored.

You can include notes to yourself in the seed file by using *comments* (preceded by a # sign) after the hostname or IP address on a line.

Example seed file:

```
node1.division.company.com
router4.division.company.com #Gateways are best
192.0.2.44
192.0.2.45
```

---

**TIP**

---

Bridges and hubs may or may not maintain ARP tables. The ARP tables are the key to expanding NNM's discovery.

2. Make a backup copy of the netmon's local registration file:

- *Windows:* `install_dir\lrf\netmon.lrf`
- *UNIX:* `$OV_LRF/netmon.lrf`

3. Load the netmon.lrf file into your text editor. The netmon.lrf file consists of the following.

```
netmon:netmon:  
OVs_YES_START:ovtopmd,pmd,ovwdb:-P:OVs_WELL_BEHAVED: 15:
```

4. Insert the seed file option (`-s filepath`) after the `-P`.

For example, adding the seed file, `nodelist`, to use as input to netmon on a management station running the Windows operating system would change the file to look like:

*Windows* (Note the required `\` character after the drive letter and the `\` character in the path statement.):

```
netmon:netmon:  
OVs_YES_START:ovtopmd,pmd,ovwdb:-P -s  
"C:\install_dir\conf\nodelist":OVs_WELL_BEHAVED:15:
```

*UNIX:*

```
netmon:netmon:  
OVs_YES_START:ovtopmd,pmd,ovwdb:-P -s  
/etc/opt/OV/share/conf/nodelist:OVs_WELL_BEHAVED:15:
```

5. Save the edited netmon.lrf file in the following location:

- *Windows:* `install_dir\lrf\netmon.lrf`
- *UNIX:* `$OV_LRF/netmon.lrf`

6. Stop the netmon service. At the command prompt, type:  
**ovstop netmon**

7. To update NNM's configuration file, `ovsuf`, at a command prompt, type:

*Windows:* **ovaddobj install\_dir\lrf\netmon.lrf**

*UNIX:* **ovaddobj \$OV\_LRF/netmon.lrf**



`ovaddobj` updates `netmon`'s information in the `ovsuf` configuration file (which instructs the start-up process what to start).

8. Restart the `netmon` service. At the command prompt, type:

```
ovstart netmon
```

NNM starts discovering from all locations identified in your seed file. Give NNM time to complete the process. Then you can manually make additions/changes to the map after the effects of the seed file are in place (see page 100).

## Use loadhosts

`loadhosts` is a program provided with NNM that allows you to place a list of IP addresses directly into the topology database. This is useful when:

- There are few SNMP nodes on your network. Using `loadhosts` speeds up the discovery process.
- You already maintain a list of the important SNMP nodes on your network.

*Windows:*

```
\WinNT\system32\drivers\etc\hosts file or  
\WinNT\system32\drivers\etc\Lmhosts file
```

*UNIX:* /etc/hosts file

Using `loadhosts` ensures that your critical devices are immediately included in NNM's databases and on the map.

- If it is critical to keep network management traffic to an absolute minimum (see page 110).

See the *loadhosts* reference page in NNM's online help (or the UNIX manpage) for more information.

1. Create an ASCII file listing each IP address or hostname on a separate line.
2. At the command prompt, type:

```
loadhosts -m subnet-mask filename
```

The following example loads the `\etc\hosts` file on a management station running a Windows operating system, with the subnet mask set:

```
loadhosts -m 255.255.248.0 \WinNT\system32\drivers\etc\hosts
```

The `netmon` service receives events about the new additions to the topology database and runs the discovery and configuration poll on those newly added nodes.

### **Expand the IPX Hop Count (Windows only)**

If you installed the Windows drivers for the IPX stack, gateways and servers acting as routers return a hop count during discovery (how many hops through gateways and routers to get to the network) for each network of which the router is aware. You can control the extent of IPX discovery allowed by NNM.

To increase the maximum hop count for discovery, open any submap and select the `Options:Network Polling Configuration IP/IPX` menu item. On the `IPX Discovery` tab, specify the hop count of your choice. This change takes effect the next time IPX discovery runs.

To discover *all* IPX networks, set the value up to 16—the maximum allowed by IPX. However, see note below.

---

#### **NOTE**

Setting the maximum hops to 16 can cause a large number of nodes to be discovered rapidly, causing both short-term and long-term system performance problems due to polling traffic. Unless you are sure your system is appropriately configured to handle the number of IPX nodes in your environment, Hewlett-Packard recommends that you expand your management domain slowly.

---

## Automatically Limit Your Management Domain

If your network maps are too complex to be useful, you have several choices to consider for customizing the discovery process.

If all the devices that are shown on your map must remain in your management domain, you can use **container** icons (see page 234) and create custom submaps to break up the visual information into smaller chunks and customize your map in a number of ways. (See Chapter 8, “Map Customization,” on page 197 for information about customizing the map itself rather than customizing the discovery process.)

If you do not need to retain all the devices that are shown on your map, NNM has options that you can use individually or in combination:

- `netmon.noDiscover` File (page 108)

You create this file listing IP addresses or ranges of IP addresses for NNM to *exclude* from the discovery process; therefore, excluding them from NNM’s databases and maps. During initial discovery when NNM discovers a device, the device’s IP address is checked against the `netmon.noDiscover` file. If the IP address is matched in the `netmon.noDiscover` file, the device is dropped and not polled for any more information. The device may be discovered again during subsequent new-node-discovery polls; however, it will once again be discarded. This is the easiest method for limiting discovery and also reduces network management traffic more than using a discovery filter (explained below).

- `loadhosts` Program (page 110)

If you need to keep network management traffic to an absolute minimum, you can turn off `netmon`’s discovery polling cycle and then use the `loadhosts` program to add your network’s critical devices directly to your topology database. Only the devices that you specify in the ASCII file of IP addresses/hostnames are included in your management domain.

---

**TIP**

---

If you already keep a local hosts file that lists important SNMP-enabled nodes on your network, you can use that file.

- Discovery Filter (page 111)

You can write Boolean logic into the NNM filter file to specify which devices will or will not pass initial discovery and be added into NNM's object and topology databases. A discovery filter is much more work for the `netmon` service than the `netmon.noDiscover` file because *each device is polled for all MIB attributes* before it is checked against the discovery filter (this is because a discovery filter can be applied to many attributes of a device). If the device does not pass the discovery filter, all information is discarded. The device may be discovered and polled again during subsequent new node discovery polls; however, the information gathered will once again be discarded.

- `oid_to_type` File (page 112)

You can open and modify the `oid_to_type` file so that specific *types* of devices are automatically configured as *unmanaged* during initial discovery. The devices show up on your map and are included in the NNM databases, but won't be included in the network management polling traffic generated by NNM.

- IPX Hop Count (Windows only, page 113)

If you installed the Windows drivers for the IPX stack, during discovery gateways and servers acting as routers return a hop count (how many hops are allowed through gateways and routers to get to the network) for each network of which the router is aware. You can restrict the extent of discovery by lowering the maximum hop count.

## Create a `netmon.noDiscover` File to Exclude Devices

The `netmon.noDiscover` file works at the very beginning of the discovery process. Entries in the `netmon.noDiscover` file are the “don't discover” list. No polling (SNMP or ICMP) is done on devices listed in the `netmon.noDiscover` file. Devices listed in your `netmon.noDiscover` file are not included in NNM's databases or displayed on NNM's maps.

If you are using HSRP, you do not need to add the virtual IP address of your HSRP-configured routers to the `$OV_CONF/netmon.noDiscover` file.

NNM does *not* attempt to discover or manage the virtual IP address of an HSRP group and does not manage the actual IP address of router interfaces in the HSRP Group. This stops NNM from deleting and rediscovering the virtual IP address in a regular pattern, which can cause participating routers to be deleted from and re-added to your map.

If you are using an existing database discovered by an older version of NNM, and NNM has already discovered the virtual IP address of your HSRP-configured routers, use the following procedure to remove the virtual IP addresses:

1. Execute `ovtopofix -r a.b.c.d` where `a.b.c.d` is the virtual IP address of your HSRP-configured routers.
2. Run `ovtopodump -lR` to make sure the virtual IP address is gone. See the `ovtopodump`, `ovtopofix`, and `netmon.noDiscover` reference page (or the UNIX manpage) for more information.

For more detailed information, see the `netmon.noDiscover` reference page in NNM's online help (or the UNIX manpage).

Use the following procedure to create entries in the `netmon.noDiscover` file.

1. Create an ASCII file. Name and place it as follows:
  - *Windows:* `install_dir\conf\netmon.noDiscover`
  - *UNIX:* `$OV_CONF/netmon.noDiscover`
2. List specific IP addresses or IP wildcards, one per line. Example `netmon.noDiscover` file entries are:

```
10.2.112.86 # one node
10.2.*.* # all nodes from 10.2.0.0 to 10.2.255.255
10.2.1-255.0-49 # first 50 nodes in 10.2.1-255 range
*.*.*.* # all nodes - nothing will be discovered
10.2.4-5.* # all nodes in either 10.2.4.* or 10.2.5.*
```
3. After you have created or modified the `netmon.noDiscover` file, force `netmon` to acknowledge the change by doing one of the following:
  - To only affect future discovery, stop and restart the `netmon` service.
  - To force changes on the map as well as future discovery:

- Stop and restart the `netmon` service.

Use the `Edit:Find` feature with the same criteria that you used in your `netmon.noDiscover` file to select all the map objects to be excluded from the future discovery. Then delete them from your map. They will not be rediscovered.

-OR-

- Wipe out the databases and start again by following the directions in “Stop Everything and Start Discovery Over Again” on page 144.

---

**TIP**

If you change your mind later and *remove* IP addresses from the `netmon.noDiscover` file, simply stop and restart the `netmon` service to force NNM to acknowledge the change. The nodes will be rediscovered at the next polling cycle.

---

## Use loadhosts

`loadhosts` is a program provided with NNM that allows you to place a list of IP addresses directly into the topology database. This is useful when:

- There are few SNMP nodes on your network. Using `loadhosts` speeds up the discovery process (see page 105).
- You already maintain a list of the important SNMP nodes on your network:

*Windows:*

`\WinNT\system32\drivers\etc\hosts` file or  
`\WinNT\system32\drivers\etc\lmhosts` file

*UNIX:* `/etc/hosts` file

Using `loadhosts` ensures that your critical devices are immediately included in NNM’s databases and on the map.

- It is critical to keep network management traffic to an absolute minimum. Turn off the new-node discovery polling process, then use the `loadhosts` program to accomplish discovery of the critical devices you wish to monitor. As long as new-node discovery is turned off, these are the only devices that will appear on your maps.

See “Controlling the Amount of Traffic Generated by NNM” on page 163 for more information about NNM’s polling cycle choices.

See the *loadhosts* reference page in NNM’s online help (or the UNIX manpage) for more information.

1. Create an ASCII file listing each IP address or hostname on a separate line.
2. Turn off NNM’s new-node discovery polling feature.

From any submap, select the Options:Network Polling Configuration menu item and disable the New-Node Discovery feature.

See NNM’s online help from this dialog box for more information.

3. If you want to start initial discovery over again, see “Stop Everything and Start Discovery Over Again” on page 144.
4. At the command prompt, type:

**loadhosts -M subnet-mask filename**

The following example loads the `\etc\hosts` file on a management station running the Windows operating system, with the subnet mask set:

```
loadhosts -m 255.255.248.0 \WinNT\system32\drivers\etc\hosts
```

The netmon service receives events about the new additions to the topology database and runs the discovery and configuration poll on those newly added nodes.

## Create a Discovery Filter Identifying Which Devices to Include

All filters consist of sets of Boolean logic that you enter into the NNM filters file:

- *Windows:* `install_dir\conf\c\filters`
- *UNIX:* `$OV_CONF/$LANG/filters`

After creating and testing your *discovery* filter, open the Options:Network Polling Configuration dialog box and enter the name of your filter in the Use filter field. Your filter will be used during all future discovery processes.

To update the object database and enforce the filter rules upon past discoveries, you must use the `ovtopofix` command to force `netmon` to acknowledge your new filter and clean out the existing information in NNM's databases so that your discovery filter changes are reflected on your maps.

See *A Guide to Scalability and Distribution* for information about writing, testing, and implementing the discovery filter. See also the *OVfilterIntro*, *netmon*, *ovfiltercheck*, and *ovfiltertest* reference pages in NNM's online help (or the UNIX manpages) for more information.

---

**TIP**

While learning how to use the Boolean language, you may wish to explore the information about *map* filters first (rather than discovery filters). NNM interactively reflects changes to the map filter on the existing maps (rather than requiring that you run the `ovtopofix` command first). Therefore, you can try different Boolean strategies and see their effect on the map right away. See *A Guide to Scalability and Distribution* for more information.

---

## Modify the `oid_to_type` File to Unmanage Devices by Type

You can instruct NNM to automatically set specific types of devices to *unmanaged* during discovery, such as all end-user PCs or all printers. If you want to use this feature *after* you have already finished initial discovery, make the changes to the `oid_to_type` file and see “Stop Everything and Start Discovery Over Again” on page 144.

The `oid_to_type` file (included with NNM) lists a large number of known `sysObjectIDs`. You can use the `U` parameter to automatically set specific types of devices to unmanaged during initial discovery. This flag only affects initial discovery status. You can change any device from unmanaged to managed at a future time.

You do not need to stop NNM when editing this file. To notify `netmon` of the changes when you are done:

- To only affect devices discovered in the future, at the command line type `nmppolling -event`



- To go back and start initial discovery again so that this change affects all devices on your network, see “Stop Everything and Start Discovery Over Again” on page 144.

See the directions at the beginning of the editable ASCII `oid_to_type` file. See also the `oid_to_type` references pages in NNM’s online help (or the UNIX manpages) for more information.

The following example entries would set devices containing either of two specific Windows versions to unmanaged, would set to unmanaged any IPX device that is not specifically identified elsewhere in the `oid_to_type` file, and would set to unmanaged any non-SNMP/IPX/IP device not specifically identified elsewhere in the `oid_to_type` file:

```
1.3.6.1.4.1.311.1.1.3.1::Windows NT:U # Windows NT
1.3.6.1.4.1.311.1.1.3.1.1::Windows NT:U # Windows NT
DEFAULT:::U
DEFAULT_IP:::
DEFAULT_IPX:::U
DEFAULT_SNMP:::
```

The `oid_to_type` file acknowledges specific `sysObjectIDs`, then wildcard `sysObjectIDs`, and then the `DEFAULT` settings.

## Limit the IPX Hop Count (Windows only)

If you installed the Windows drivers for the IPX stack, gateways and servers acting as routers return a hop count during discovery (how many hops through gateways and routers to get to the network) for each network of which the router is aware. You can control the extent of IPX discovery allowed by NNM.

To reduce the maximum hop count for discovery, open any submap and select the `Options:Network Polling Configuration IP/IPX` menu item. On the `IPX Discovery` tab, specify the hop count of your choice.

The extra devices that were discovered based upon the old hop-count number must be manually deleted from your map. Once deleted, the discovery process will not add them again.

## Troubleshooting Discovery

This section contains information about resolving common challenges encountered during NNM's discovery of your network:

- IP discovery problems and solutions (page 114)
- IPX discovery problems and solutions (page 136)

Often times, the trouble is caused because of breakdowns in your well-configured network plan (see “Starting with a Well-Configured Network” on page 72): GET and SET community names other than *public*, DNS name resolution inconsistencies, duplicate IP addresses, incorrect subnet masks, or incorrectly configured SNMP agents or DMI service providers on specific devices. NNM's map and event tracking system can make it much easier for you to identify offending devices and isolate what exactly is wrong, so you can fix the problem.

Remember to take advantage of the online discussion group, OV Forum, provided for NNM customers to communicate with each other (see “OpenView Forum” on page 59). You will find a wealth of information there.

### IP Discovery and Layout

The IP troubleshooting information is broken into several groups:

- “Network Issues” on page 116
  - “Opening NNM takes a long time” on page 116
  - “NNM's initial discovery found zero nodes” on page 116
  - “All or most of the map is wrong” on page 118
  - “Whole IP networks are missing from my map” on page 118
  - “Strange networks appeared on my map” on page 119
- “Router Issues” on page 119
  - “My default router shows up on the map as a host” on page 119
  - “Some of my routers are missing” on page 119

- “One router shows up twice on my map as if it were two devices” on page 120
- “Logical interfaces (secondary IP addresses) are missing from my router’s interface card submap” on page 120
- “NNM’s discovery process negatively impacts my router’s performance” on page 121
- “My routers are set to managed but the map doesn’t expand” on page 121
- “Problems With Specific Devices” on page 121
  - “IP addresses are used to identify devices, rather than their hostnames” on page 121.
  - “The map includes devices and routing links that I have never heard of before” on page 122
  - “A few devices are depicted inaccurately on the map” on page 122
  - “An SNMP device shows up on the map as a non-SNMP node” on page 122
  - “A host shows up on the map as a router” on page 123
  - “Some of my devices are missing from the map” on page 124
  - “Devices are labeled with the IP address, when I have a name resolution system in place. What happened to the hostname” on page 124
  - “Multi-homed hosts show up as separate devices” on page 125
  - “How does NNM decide which one of multiple hostnames to use for a device” on page 125
  - “Devices have generic icons rather than manufacturer-specific icons” on page 125
  - “The name of a system running Windows is incorrect on the map” on page 126
  - “Addresses from two distinct nodes within a system running Windows are merged into a single node” on page 126
  - “DNS hostname is incorrect on map symbols for a system running Windows” on page 126
- “General IP Suggestions” on page 127

- “Check NNM’s Alarm Browser: Configuration Alarms” on page 127
- “Duplicate IP Address Errors” on page 127
- “Routing Table Configuration” on page 127
- “GET- and SET-Community Name and SNMP Port Issues” on page 128
- “Subnet Mask Issues” on page 132
- “Running netmon with the Jumpstart (-J) option” on page 135

## Network Issues

### Opening NNM takes a long time

This is probably caused by an incorrectly configured name resolution scheme. To verify that this is the problem, do the following:

1. Find the `checkDNS.ovpl` tool.

*Windows:* (located on the NNM installation CD)  
`\support\checkDNS.ovpl`

You must copy `checkDNS.ovpl` and `gethost` into the following directory on the NNM management station:

`install_dir\support\*`

*UNIX:* `/opt/OV/support/checkDNS.ovpl`

2. Navigate to the directory containing `checkDNS.ovpl` on the NNM management station. At the command prompt type:

**`checkDNS.ovpl -v`**

3. Every device listed in the NNM topology database is issued an appropriate `gethostbyaddr` or `gethostbyname` system call. Addresses or hostnames that are too slow in responding are identified in the output on the screen. Clean up the name resolution for addresses or hostnames identified as being too slow.

### NNM’s initial discovery found zero nodes

- First, verify that NNM’s `netmon` service is running:
  - *Windows:* Select `Start:Programs:HP OpenView: Network Node Manager Admin->NNM Services - Status`.

— *UNIX*: At the command line prompt, type `ovstatus -c`.

If any of the services are not successfully running, see “Troubleshooting NNM Itself” on page 537 for information about troubleshooting services before continuing.

- Make sure that your network devices are running and responding to pings and SNMP requests. From the command line, issue the ping commands and the SNMP requests (`snmpget` and `snmpwalk`), or from any submap use the `Fault:Network Connectivity` menu choices.

Make sure that SNMP agents are installed and running on each agent system, particularly on gateways. The problem may be that there are no SNMP agents installed.

Your SNMP GET- and SET-Community names may not be matching up with the settings on the management station. See “GET- and SET-Community Name and SNMP Port Issues” on page 128 for more information.

- If you have very few SNMP nodes on your network, run `loadhosts` to force `netmon` to find them. See “Use loadhosts” on page 105.

Or run `netmon` with the `jumpstart (-J)` option. See “Running `netmon` with the Jumpstart (-J) option” on page 135 for more information.

- Make sure that the appropriate SNMP agent is installed and running on the management station:

— *Windows*: `SNMP EMANATE Master Agent Service`. The service must be running on the management station for discovery to function correctly. You can verify status in the `Windows NT Services` applet in the Control Panel.

— *UNIX*: `snmpd`. An `snmpd` agent must be running on the management station for discovery to function correctly.

- If you specified a seed file, verify that the seed file exists and that you are using the seed file’s correct name in `netmon`’s LRF file. Also, verify that you included 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. Do not use bridges and hubs as seeds, since they do not maintain ARP tables.
- From any submap, select the `Fault:Network Connectivity:Ping` menu item to contact your default router and the closest SNMP-supporting router to start gathering ARP table information.

- Make sure your default route is configured correctly. This information determines how IP packets are routed on nodes. To find the default route, do one of the following:
  - Select your management station’s symbol on the NNM map and select the Configuration:Network Configuration:IP Routing Table menu item.

---

**TIP**

---

You may want to establish a policy for using a standard default route so that only one router needs to maintain the routing table.

- From the command prompt, issue the following command:
  - *Windows:* ipconfig /all
  - *UNIX:* netstat -r

### **All or most of the map is wrong**

If all or most of the map database is inaccurate, do the following.

On your NNM map, select the questionable router symbols on the map, then use the Edit:Object Properties and select IP Map to see the current subnet mask configuration.

If a router’s subnet masks are incorrectly configured, correct the configuration (see “Subnet Mask Issues” on page 132) and restart discovery (see “Stop Everything and Start Discovery Over Again” on page 144).

### **Whole IP networks are missing from my map**

NNM isn’t yet managing any devices within the missing IP network:

- Routers with addresses in the missing IP networks may not be accessible via SNMP. See “GET- and SET-Community Name and SNMP Port Issues” on page 128.
- See “Interactively Expand/Limit Your Management Domain” on page 100.
- If you have several additional networks that you need to add, consider creating a seed file to set devices within those networks to *managed* all at once. See “Create a Seed File Specifying Multiple IP Networks to Manage” on page 103 for more information.

### Strange networks appeared on my map

The most common cause of this problem is incorrect subnet masks. See “Subnet Mask Issues” on page 132 for more information.

To check the network device’s current subnet mask configuration, select the device’s symbol on the map and use the map’s `Edit:Object Properties` and select `IP Map`.

### Router Issues

#### My default router shows up on the map as a host

If the default router shows up as a host on the map, verify the following:

- The most common reason that `netmon` incorrectly identifies a router as a host is that your `SNMP GET-` and `SET-Community` names may not be matching up with the settings on the management station. See “`GET-` and `SET-Community Name and SNMP Port Issues`” on page 128 for more information.
- Make sure that `IP Forwarding` on the router is set to “forwarding” *and* that the router has more than one interface card. From any submap, select the router’s symbol and select the `Tools:SNMP MIB Browser` menu item and browse the following information:

— `IP Forwarding configuration (.1.3.6.1.2.1.4.1)`

`.iso.org.dod.internet.mgmt.mib-2.ip.ipForwarding`

— `Number of interfaces (.1.3.6.1.2.1.2.1)`

`.iso.org.dod.internet.mgmt.mib-2.interfaces.ifNumber`

`IP Forwarding` is a vendor/device-specific configuration. See your device’s documentation for more information.

#### Some of my routers are missing

If routers are not showing up on your network, check to see if the routers are “backup” routers; that is, routers used in an emergency when another router goes down. `netmon` may not discover backup routers, because `netmon` looks only for nodes on the network that are actually used. Therefore, a backup router will be discovered either as a host or it may not be discovered at all. To solve this problem, follow these steps:

1. Manually add the router to the map. See “Interactively Expand/Limit Your Management Domain” on page 100 for information about manually adding the router.
2. Use the `Fault:Network Connectivity:Ping` to determine the initial status condition of the router and turn its map symbol green (managed).
3. Use the `Fault:Network Connectivity:Poll Node` to get the information that `netmon` needs to connect the router in the proper place on the map.

### **One router shows up twice on my map as if it were two devices**

See “Multi-homed hosts show up as separate devices” on page 125.

### **Logical interfaces (secondary IP addresses) are missing from my router’s interface card submap**

Secondary addresses are logical interfaces that are configured in addition to the physical interfaces on a router. Unfortunately, for some agents, these additional interfaces are not included in the IP address table returned through SNMP. Therefore, NNM is unaware that these interfaces exist.

When an IP address is found in an ARP Cache and it resolves to the same official name as a router, the IP address is added as an additional interface. When the router’s node submap is displayed, the primary interfaces (the ones returned through SNMP) have labels that contain the first word of the interface description. The secondary addresses (the ones discovered in an ARP Cache) have only their IP addresses as their labels.

NNM can discover secondary addresses only if:

- The secondary address resolves to the same official name as the router.
- The secondary address is connected to an existing subnet.

If NNM fails to discover all the configured secondary addresses, they can be manually added as interface symbols to the router’s node submap. If the interface is connected to a new subnet, that subnet must also be manually added. NNM will not automatically add the subnet symbol. See the `netmon` reference page in NNM’s online help (or the UNIX manpage) for information about the `-S` parameter.



The secondary addresses can be placed in a seed file that will be used by the `netmon` service for discovery (see “Create a Seed File Specifying Multiple IP Networks to Manage” on page 103). In this case, the secondary address interface symbols and the corresponding networks will be added to the map automatically.

### **NNM’s discovery process negatively impacts my router’s performance**

This can happen when NNM is requesting ARP cache tables or routing tables from the router. You can use the `netmon -R` parameter to stop NNM from polling any routing tables. This cuts down on polling traffic since routing tables can be quite extensive. However, using this option means that NNM cannot discover remote routers over serial WAN links without the help of either a seed file (page 103) or loadhosts (page 105). See the `netmon` reference page in NNM’s online help (or the UNIX manpage) for more information about `netmon -R`.

### **My routers are set to managed but the map doesn’t expand**

One of the following may be the problem:

- Open the Interface submap for each router by double-clicking on the router symbol and verify that all interface cards are discovered. If any are missing add them manually.
- The missing network may simply be a quiet network; without network traffic, the ARP tables won’t be populated to aid NNM in the discovery process. See “Running `netmon` with the Jumpstart (-J) option” on page 135.
- Your router may have static routes over serial interfaces. NNM can’t discover the information it needs to expand discovery because static routes may not have an entry in the routing table with a next hop address. To bypass this issue:
  - *Windows*: Manually add the corresponding interface card.
  - *UNIX*: Ping the corresponding interface card.

### **Problems With Specific Devices**

#### **IP addresses are used to identify devices, rather than their hostnames**

See “Opening NNM takes a long time” on page 116. See “Consistent Name Resolution Scheme” on page 73 for more information.

### **The map includes devices and routing links that I have never heard of before**

This is probably caused by incorrectly configured subnet masks. See “Subnet Mask Issues” on page 132 for information about fixing problems with subnet masks.

### **A few devices are depicted inaccurately on the map**

If isolated information in the map database is inaccurate, check the following:

- From any submap, select the inaccurate symbol, right-click with the mouse, and select *Object Properties, IP Map*. Write down the *System Object ID* number. Verify that the device supports SNMP (*isSNMPsupported=true*). Then, open the *oid\_to\_type* file and check the files in the *oid\_to\_sym\_reg* directory structure to make sure there is an appropriate entry for that *sysObjectID* number. Also, check the registration files included in the *install\_dir\symbols\(\$OV\_SYMBOLS/\$LANG)* directory. Make any additions or corrections. Then, select the node’s symbol on any submap, select *Fault:Network Connectivity->Capability Poll*. It may take some time before the map symbol updates. See Appendix C, “Changing All the Symbols for a Particular Device,” on page 623 for more information.
- If changes to the underlying object’s configuration have recently been made, select the device’s symbol on any submap, then select the *Fault:Network Connectivity:Poll Node* menu item. This forces an immediate configuration check and updates NNM’s databases to reflect your changes. You can also simply wait until the next polling cycle.

### **An SNMP device shows up on the map as a non-SNMP node**

If NNM incorrectly identifies an SNMP node as a non-SNMP node, therefore displaying it as an unmanaged symbol:

- The problem may be that all of *netmon*’s SNMP requests to a node failed. During SNMP requests, *netmon* asks for system information, interface table, IP address table, routing table, ARP cache, and possibly Bridge or Repeater MIB tables.

If the node does not reply with a `sysObjectID`, `netmon` assumes that the node does not support SNMP. If, however, a configuration check succeeds, `netmon` assumes the node *does* support SNMP. Note that SNMP requests can fail if the transport is very busy. SNMP runs over UDP.

- The most common cause of this problem is that your SNMP GET- and SET-Community names may not be matching up with the settings on the management station. See “GET- and SET-Community Name and SNMP Port Issues” on page 128 for more information.
- Another common reason for failures is that the SNMP agent returns incorrect information. See “GET- and SET-Community Name and SNMP Port Issues” on page 128 for more information.

### A host shows up on the map as a router

If you have a host that shows up as a router, it may be because the host has *multiple interfaces* and *IP forwarding*, therefore meeting the criteria of a router. To check the current configuration on the host:

1. From any submap, select the host’s symbol and select the Tools:SNMP MIB Browser menu item and browse the following information:
  - IP Forwarding configuration (.1.3.6.1.2.1.4.1)  
.iso.org.dod.internet.mgmt.mib-2.ip.ipForwarding
  - Number of interfaces (.1.3.6.1.2.1.2.1)  
.iso.org.dod.internet.mgmt.mib-2.interfaces.ifNumber
2. From any submap, select the host’s symbol and determine the device’s System Object ID (`sysObjectID`) by using Edit:Object Properties and selecting IP Map. Use the Edit:Find->Object By Attribute menu item.

Find all devices with the same `sysObjectID`.

If all instances of this device are showing the same misbehavior on the map, see the *oid\_to\_type* reference page in NNM’s online help (or the UNIX manpage) for information about the `M` parameter in the *oid\_to\_type* file that designates a device as a *multi-homed host* (not a gateway).

### Some of my devices are missing from the map

There may be a problem with ICMP communication (ping):

- Select the `Fault:Network Connectivity:Ping` menu item and enter the missing IP address. If you can communicate directly with the device in this manner, it will automatically be added to the map.
- See “Interactively Expand/Limit Your Management Domain” on page 100.
- Check the device itself to ensure that it is not powered down.

The device may be in a network which is not yet *managed* by NNM:

- See “Interactively Expand/Limit Your Management Domain” on page 100.
- If you have several additional networks you need to add, consider creating a seed file to set multiple networks to *managed* all at once. See “Create a Seed File Specifying Multiple IP Networks to Manage” on page 103 for more information.

If initial discovery doesn’t find many SNMP objects or if your SNMP community names are in disarray, the `netmon jumpstart` command (`-J` option) can force detected SNMP agents to generate ICMP broadcasts to populate their ARP tables, thus aiding the discovery process. See “Running netmon with the Jumpstart (-J) option” on page 135 for more information.

### Devices are labeled with the IP address, when I have a name resolution system in place. What happened to the hostname

Check the DNS, NIS, or other name resolution configuration.

Use the `nslookup` tool that is included in the Windows or UNIX operating system. This utility can easily verify which part of the name-resolution configuration is incorrect. The reverse map must also work (`nslookup` must be able to resolve an IP address to a hostname, as well as resolve a hostname to an IP address). See your operating system’s documentation for more information.

---

#### TIP

Make sure that the case (use of uppercase and lowercase letters) is consistent in your hostnames.

---

### Multi-homed hosts show up as separate devices

- The most common reason that `netmon` incorrectly identifies a multi-homed host as separate devices is that your SNMP GET- and SET-Community names may not be matching up with the settings on the management station. See “GET- and SET-Community Name and SNMP Port Issues” on page 128 for more information.
- Check the DNS, NIS, or other name resolution configuration.  
See also “Devices are labeled with the IP address, when I have a name resolution system in place. What happened to the hostname” on page 124 and “The name of a system running Windows is incorrect on the map” on page 126 for more information about name resolution.
- See the `oid_to_type` reference page in NNM’s online help (or the UNIX manpage) and explore the `M` parameter setting for multi-homes hosts.

### How does NNM decide which one of multiple hostnames to use for a device

NNM uses the hostname associated with the lowest numbered IP address which resolves to a name.

### Devices have generic icons rather than manufacturer-specific icons

- The most common reason that devices display on the map with generic icons is that your SNMP GET- and SET-Community names may not be matching up with the settings on the management station. See “GET- and SET-Community Name and SNMP Port Issues” on page 128 for more information.
- From any submap, select the inaccurate symbol, right-click with the mouse, and select `Object Properties, IP Map`. Verify that the device supports SNMP (`isSNMPsupported=true`). Write down the `System Object ID` number. Then, open the `oid_to_type` file and check the files beneath the `oid_to_sym_reg` directory structure to make sure there is an appropriate entry for that `sysObjectID` number. Also, check the registration files included in the `install_dir\symbols\($OV_SYMBOLS/$LANG)` directory. Make any additions or corrections. Then, select the node’s symbol on any submap, select `Fault:Network Connectivity->Capability Poll`.

It may take some time before the map symbol updates. See Appendix C, “Changing All the Symbols for a Particular Device,” on page 623 for more information.

**The name of a system running Windows is incorrect on the map**

If an IP address to IP hostname lookup fails, the system attempts to resolve the address via NETBIOS. If the system with the IP address does not have a NETBIOS name that can be resolved back to the IP address in question, the symptoms described above can occur, as well as other communication problems with the target system.

To verify and correct the NETBIOS name on the target system, do the following:

1. On the remote system, access the Windows Control Panel.
2. Double-click on the Network icon; a dialog box will appear.
3. Correct the configured computer name.

---

**NOTE**

Changing the name of the target system may affect other network connections for the system.

---

See also “Devices are labeled with the IP address, when I have a name resolution system in place. What happened to the hostname” on page 124 for more information about name resolution.

**Addresses from two distinct nodes within a system running Windows are merged into a single node**

See “The name of a system running Windows is incorrect on the map” on page 126 for information about resolving this.

**DNS hostname is incorrect on map symbols for a system running Windows**

See “The name of a system running Windows is incorrect on the map” on page 126 for information about resolving this.

## General IP Suggestions

### Check NNM's Alarm Browser: Configuration Alarms

NNM's Alarm Browser sorts network alarms into useful categories for you. With your map open, notice the Alarm Categories window off to the side of your map. Click on the Configuration Alarms button to display the list of problems that NNM has discovered. Fix any problems before continuing. You will find information such as:

- Inconsistent subnet masks discovered on specific nodes (see “Subnet Mask Issues” on page 132).
- Duplicate IP addresses discovered on specific nodes.

### Duplicate IP Address Errors

Duplicate IP addresses may cause the following NNM behavior:

- In an effort to keep the map current, NNM may continuously move the device symbol from one submap to a second submap and back again as it discovers the first device using the IP address and then discovers the second device using the same IP address.
- NNM's Alarm Browser may show multiple Mismatched link-level address alarms indicating that ARP tables on your network devices are continuously working to update based upon messages received from the two devices using the same IP address.

If within the alarm message NNM identifies the host name of the two devices, correct the problem on one of the two offending devices.

If within the alarm message NNM identifies the MAC address or link-level address of the two devices, use NNM's Edit:Find->Object by attribute feature to identify the two offending devices.

### Routing Table Configuration

Make sure your network device's routing tables are configured properly. To check the routing table configuration on an SNMP-enabled device, select the device's symbol on any submap, then select Configuration:Network Configuration->IP Routing Table.

---

**TIP**

NNM does not detect proxy ARP (an alternative to defining extensive routing tables). If you receive alarms about misconfigured subnet masks, proxy ARP may be the problem. If you wish to prevent these alarms from posting, highlight one of the alarms in the Alarm Browser and select Action:Configure Event. Then modify the event's configuration so that the alarm category is set to Don't log or display.

---

### **GET- and SET-Community Name and SNMP Port Issues**

Community names are a security technique employed by SNMP agent software. If the network management software wants to communicate with the SNMP agent software, both the management station and the SNMP agent must know the current GET-community name and talk over the same port. By default, community names are often set to *public* and the port is set to 161. NNM assumes that these defaults are in use unless you inform it otherwise. Many network administrators change the GET-Community name so that hackers cannot obtain sensitive information about their devices. It is even more common for network administrators to change the SET-Community name, since knowing this community name enables others to make system configuration changes to the device itself.

In order to work, NNM depends upon successful communication with the SNMP agents installed on your network devices. If the GET-community names have been changed from *public* or the port changed from 161, you need to inform NNM of the new ones. (NNM cannot obtain these automatically because that would defeat the security feature.)

---

**NOTE**

If you do not have access to the GET-Community names, for critical devices in your network, see "If Community Names are Unavailable" on page 131.

---

To inform NNM of the community naming scheme and ports in use on your network:

1. Open any submap and select the Options:SNMP Configuration menu item.



2. Enter the specific information about your community naming scheme for IP and IPX devices in this dialog box. For your convenience, you can use wildcard entries when indicating IP addresses. If no SET-Community name is defined, NNM uses the GET-Community name for SNMP SetRequests.

In this same dialog box, you can indicate any SNMP proxies that you have set up for non-SNMP devices and indicate any nonstandard remote-port configurations that are in use by SNMP agent software on your network (default is port 161). Generally, a remote port is only specified for specialized proxy agents that do not listen to the standard SNMP port.

Check NNM's online help from this dialog box for more information and directions.

For technical information, see the *ovsnmp.conf* reference page in NNM's online help (or the UNIX manpage).

You can also change these values using a command line interface. For more information, see the *xnmsnmpconf* reference page in NNM's online help (or the UNIX manpage). If you need to know the specific configuration settings for a particular device, use:

```
xnmsnmpconf -resolve device_name
```

If you purchased the NNM Advanced Edition and need information about configuring SNMP for collection stations and management stations, refer to *A Guide to Scalability and Distribution*.

To determine which GET- and SET-Community name is currently configured in the SNMP agent on each device in your network, refer to the documentation that came with each SNMP agent software package.

If you change the community names on the agent, you also *must* configure the community names for that agent on the management station.

If the settings on both your management station and your SNMP device's agent software are correct and it still does not work, try the following and then contact the vendor of the SNMP agent software.

To verify that the problem resides in the device's SNMP agent, do one of the following. Then see if the information returned to you by the device's SNMP agent makes sense. Make sure common entries in these two tables match:

- On any submap, select the device's symbol and select the Tools:SNMP MIB Browser menu item. Then browse the following MIB tables:
  - interface table .1.3.6.1.2.1.2.2  
(.iso.org.dod.internet.mgmt.mib-2.interfaces)
  - IP address table .1.3.6.1.2.1.4.20.1  
(.iso.org.dod.internet.mgmt.mib-2.ip.ipAddrTable)
- From the command prompt using `snmpget` or `snmpwalk`, type:

```
snmpwalk nodename (or IP address) .1.3.6.1.2.1.2.2  
snmpwalk nodename (or IP address) .1.3.6.1.2.1.4.20.1
```

Contact the vendor who supplied the SNMP agent software if the information returned to your request doesn't make sense.

Symptoms of incorrect community names include

- The default router is not discovered.
- Hosts appear to not support SNMP.
- `netmon` incorrectly identifies a router as a host.
- Node that supports SNMP as not supporting SNMP.
- The default router is shown as a host on the map.
- Error messages appear, such as "time-out" or "no such name."
- `netmon` incorrectly identifies a node that supports SNMP as not supporting SNMP.
- Multi-homed hosts show up as separate devices.

If the community name on the default router or the community name on nodes in the seed file does not match the configuration settings in NNM, `netmon` will not be able to get the information that it needs in order to discover the nodes on the network.

---

**TIP**

If NNM attempts to contact a node using SNMP and the SNMP request times out, NNM looks for a `netmon.cmstr` file that you can configure to contain additional community names for the node. If the `netmon.cmstr` file exists, and contains community names, the first community name from the `netmon.cmstr` file that results in a successful SNMP response is

stored in the SNMP configuration database for that node. See the *netmon.cmstr* reference page in NNM's online help (or the UNIX manpage) for more information.

---

### If Community Names are Unavailable

If you need to manage critical devices, yet do not have access to their GET-community names, create a *hosts* file. This *optional* file allows NNM to identify multiple IP addresses as being within a single device (without obtaining this information from the device's SNMP agent). This file must be maintained manually, in the following format:

```
IPAddress      hostname      alias(optional)
```

If you are identifying multi-homed devices such as switches, you must implement alias assignments so that the hostnames of all boards within the device resolve to the same IP address.

```
IPAddress_1      hostname_1      hostname_2
IPAddress_2      hostname_1
IPAddress_3      hostname_1      hostname_3
```

Place your new *hosts* file in the following location. Make sure that the file does not have any extension, such as *.txt*:

- *Windows*: `\WinNT\system32\drivers\etc\hosts`
- *UNIX*: `/etc/hosts`

---

#### NOTE

Name resolution order issues:

*Windows*: The default behavior is to search the *hosts* file before checking DNS. If you have altered this default order, this technique may not work.

*UNIX*: Ensure that the hostname resolution order reads your new *hosts* file. Open the `/etc/nsswitch.conf` file and verify the contents.

---

## Subnet Mask Issues

If the subnet masks on the management station or configured on your network devices are incorrect, discovery will fail. If the subnet mask is too restrictive, not enough is discovered. If the subnet mask is not restrictive enough, too much is discovered.

Symptoms of incorrect subnet masks may include:

- `netmon` discovers and adds nodes from the internet that are outside of your administrative domain. This is caused by an un-subnetted or incorrectly subnetted network.
- Nodes in your management domain do not show up on the map. This is caused by incorrect subnet masks on those nodes.
- Strange networks appear on the map, such as a network having a Class A, Class B, or Class C network number when you expect a subnetted network number. This is caused by incorrectly configured subnet masks on the nodes mapped to those strange networks.
- Many `Inconsistent subnet mask` alarms show up in your Alarm Browser. This is caused by incorrect subnet masks on the nodes identified in the alarm messages.

For networks to be discovered correctly, subnet masks must be correctly configured on all IP interfaces on the following:

- The management station
- All gateways
- Any nodes in a seed file for `netmon`
- Any SNMP-enabled node within your management domain

---

### NOTE

Non-contiguous subnet masks are not supported by NNM. Also, in hierarchical subnets, two subnets having the same IP subnet address but different subnet masks are not supported. See the white paper about subnet masks (page 57) for more information.

Proxy ARP (an alternative to defining extensive routing tables) is also not supported and will result in “misconfigured subnet mask” alarms. If you wish to prevent these alarms from posting, highlight one of the

alarms in the Alarm Browser and select Action:Configure Event. Then modify the event's configuration so that the alarm category is set to Don't log or display.

---

To identify the source of the problem caused by incorrect subnet masks, first check the management station to see which subnet masks it knows about, then check the subnet configuration on each network and each device that is causing the problem.

1. *Check the subnet mask information on the Management Station:*

Check this information through both SNMP and non-SNMP tools to check for misconfigurations.

- SNMP check:

Use the Edit:Find->Object by Selection Name to locate the management station's symbol.

Select the NNM management station's symbol and select Configuration:Network Configuration->IP Addresses to see which subnet masks are in use on the management station's interfaces.

- Non-SNMP check using the tool provided by your operating system:

— *Windows:* issue the `ipconfig /all` command

`ipconfig /all` tells you how the system is actually configured; which may not be what the device's SNMP agent returns. For information, see your operating system documentation.

— *UNIX:* issue the `ifconfig interface` command, where *interface* is usually `lan0`, `lan1`, and so forth. The `interface` command can be found in the `/usr/sbin/ifconfig` file.

2. *Check the subnet mask for each network:* If the subnet masks on the management system are correct (from the previous step), determine the subnet mask currently being used for each network. Do one of the following:

## Troubleshooting Discovery

- Select a network symbol on any submap, then right-click on the symbol and select **Object Properties, IP Map**. Check to see that the subnet mask entry is correct. Repeat for each network symbol in question.
  - From the command prompt, issue the `ovttopodump` command specifying each network. See the *ovttopodump* reference page in NNM's online help (or the UNIX manpage) for more information.
3. *Check the subnet mask on individual network devices:* To determine what subnet masks are returned to NNM from a device's SNMP agent, do one of the following:
- On any node-level submap, select an interface board symbol, right-click on the symbol and select **Object Properties, IP Map**. Check the subnet mask entry.
  - On any submap, select the **Fault:Network Connectivity->Poll Node** menu item and type in the offending IP address and look at the results.
  - From the command prompt, type:  
**netstat -I nodename (or IP address)**  
  
Check the Windows operating system documentation or the UNIX manpage for more information about `netstat`.
4. To solve the subnet masks problems, follow these steps:
- a. Correct the problem with the subnet masks on the network objects that have the problem.  
  
Log into the network device itself and modify its network configuration to specify the correct subnet masks.
  - b. Use the **Fault:Network Connectivity->Poll Node** menu item to verify that the new symbols have the correct subnet masks.

---

### TIP

If you discover a large number of inaccuracies, correct the problems and then redo discovery to reset NNM's databases ("Stop Everything and Start Discovery Over Again" on page 144).

---

### Running netmon with the Jumpstart (-J) option

The `netmon` jumpstart command (-J option) can force detected SNMP agents to generate ICMP broadcasts to populate their ARP tables, thus aiding NNM's discovery process. See the *netmon* reference page in NNM's online help (or the UNIX manpage) for more information.

---

**CAUTION**

---

Generating ICMP broadcasts can cause a great deal of network traffic. Use this with caution.

`netmon`'s default setting does not use the -J option. By default `netmon` generates, if necessary, just an ARP Request. There are several factors that determine whether `netmon` needs to generate an ARP Request:

- For nodes that are up but not on the same subnet with the management station, `netmon` generates just one ICMP echo request packet and gets back one response. In this case, no ARP Request is generated.

Because the management station is on a different subnet, a request will go through a gateway. Gateways always have large ARP caches. `netmon` knows the gateway's link-level address and does not have to broadcast an ARP Request to that gateway. `netmon` only has to send one ICMP echo request through the gateway and gets back one response.

- For devices that are on the same subnet as the management station, `netmon` may generate just one ICMP echo request, or it may also generate one ARP Request as well. Because the nodes are on the same subnet, `netmon` does not have to go through a gateway. Broadcasting an ARP Request will depend on whether or not `netmon` already has the device's link-level address in the management station's own ARP cache.
- `netmon` will generate an ARP Request if the address that is being polled for status is not in the ARP cache of your management station.

The presence of the address in the management station's ARP cache is affected by how many nodes are on the subnet and by how much physical memory is in your management station (some systems will automatically adjust the size of their ARP table).

If `netmon` is started with the `-J` option, `netmon` may cause a broadcast ICMP echo request on each subnet it discovers. These broadcasts will only be issued on a subnet if:

- No broadcast ICMP echo request has ever been issued on this subnet on behalf of this management station.
- An agent capable of issuing a broadcast ICMP echo request on `netmon`'s behalf has been discovered on this subnet. All HP-UX SNMP agents have this capability.

## **IPX Discovery and Layout (Windows only)**

- “General IPX Suggestions” on page 136
- “Network Issues with IPX” on page 137
  - “All IPX nodes are missing from my map” on page 137
  - “IP Network 0.0.0.0 Contains All IPX Nodes” on page 139
- “Router Issues with IPX” on page 139
  - “IPX routers are missing from my map” on page 139
- “Problems with Specific IPX Devices” on page 140
  - “Some IPX nodes are missing from my map” on page 140
  - “IPX nodes have incorrect names or labels” on page 140
  - “Nodes supporting both IP and IPX appear on the map as two separate nodes” on page 142
  - “Duplicate IP Address messages are in the Configuration Alarms list” on page 142

### **General IPX Suggestions**

- Ensure that IPX is configured and installed correctly on the management station. See “Installing IPX Transport Software (Windows only)” on page 82. Pay particular attention to the Frame Type. All IPX nodes on the same IPX network must use the Frame Type configured on the IPX servers and routers.
- Ensure that nodes are running and responding to IPX diagnostic requests *and* SNMP requests. From the command prompt, issue the `ixping -b` command. You should receive a response from each



device configured to respond to IPX. If you don't receive any responses, your management station's IPX is probably not configured correctly. If a particular node does not respond, use `snmpget` and `snmpwalk` to check that device's configuration settings. Correct any problems that you discover.

- Ensure that the community name is configured correctly (see “GET- and SET-Community Name and SNMP Port Issues” on page 128).

It may be necessary to add entries for the IP address and hostname in addition to the IPX address. The format for an IPX address is `<net>:<nodeaddr>`; for example, `00000100:080009ABCDEF`. You can drop the leading zeros of the `<net>` portion of the address in most cases. For example: `100:080009ABCDEF`.

- Ensure that the SNMP agent on the node is running.
- Ensure that the IPX objects are *managed* on your map.

## Network Issues with IPX

**All IPX nodes are missing from my map** If NNM does not discover any IPX nodes when run for the first time, check the following:

- Make sure that IPX discovery is enabled via the `Options:Network Polling Configuration:IP/IPX operation`. Click the `IPX Discovery` tab and make sure that `Discover New Nodes` is checked. Also, if IPX discovery is configured for a particular time of day, no nodes will be discovered until that time.
- Make sure that the IPX transport is correctly configured and installed on your system. If `netmon` is started with IPX discovery enabled, but the IPX software is not installed, check the `Alarm Browser` window under the `Application Alert Alarms` category. You should see an application error event similar to `Could not get IPX RIP socket, disabling IPX discovery`.

`netmon` will not perform IPX discovery until IPX is correctly configured and `netmon` is restarted.

- Check the `Alarm Browser` window under the `Application Alert Alarms` category for other errors that may indicate the problem. For example, an invalid license could prevent the system from starting.
- Verify that there is at least one server or router that will respond to the IPX RIP request.

1. Issue the following command at the command prompt:

*Windows:* **`install_dir\bin\ipxping -r -n 1 -d 5`**

This command sends one IPX RIP request and waits five seconds for responses. Responses will include the address of the responding server or router. Adding the `-v` option will include the IPX networks that can be reached from your management station.

If you do not see a response and the management station is not also configured as an IPX/NetWare gateway, this indicates that there are either no servers or routers attached to this network, or there is a misconfiguration in your IPX Transport installation. In particular, verify that the IPX transport Frame Type is configured correctly, both on the management system and on the various servers and routers in the network.

All nodes configured for the same IPX network must run the same Frame Type. For some installations, different IPX networks may be configured for different Frame Types on the same physical cable, but all nodes in the same IPX network must have the same Frame Type, which must match the types configured on the NetWare servers and IPX routers.

2. If the test above succeeded in producing RIP responses, verify that your system receives IPX diagnostic responses by issuing the following command:

*Windows:* **`install_dir\bin\ipxping -n 1`**

This command sends a single IPX diagnostic broadcast request to the local network. All NetWare clients and servers attached to the local network should respond. Lack of response indicates either there are no local NetWare clients, or there is a misconfiguration of the IPX transport on the management station or the target systems. Verify the IPX transport installation and configuration.

3. You can also verify communication with any given network by issuing the following command from the command prompt:

*Windows:*  
**`install_dir\bin\ipxping -n 1 -b <IPX_net_addr>`**

For example: `bin\ipxping -n 1 -b 00000100` sends a single IPX broadcast diagnostic request to the IPX network 00000100. All NetWare clients and servers configured for the 00000100 network should respond.

**IP Network 0.0.0.0 Contains All IPX Nodes** If you are running this version of NNM on a collection station that forwards information to a management station running NNM 4.X, you may encounter errors on the management station's display of network information. NNM 4.X does not correctly handle IPX nodes and networks. All nodes with IPX interfaces end up in a single IP network with the IP address 0.0.0.0.

To avoid this, install a topology filter that includes *only* IP objects. You can find details on installing topology filters in *A Guide to Scalability and Distribution*. An example filter, `IPObjectsOnly`, is shipped with the system's filter configuration file for this purpose.

## Router Issues with IPX

### IPX routers are missing from my map

1. Verify connectivity with the router as described in “Some IPX nodes are missing from my map” on page 140.
2. If the router does not support IPX diagnostic protocols, you can expand the IPX discovery region beyond the router using the `Options:Network Polling Configuration` operation. Click the `IPX Discovery` tab, and use the controls to set the `Maximum Hops` to a larger value. Make sure the value is large enough to expand beyond the router, but not any larger than desired for your discovery purposes. To discover all IPX networks, set the value to 16—the maximum allowed by IPX (however, see note below).

---

#### NOTE

Setting the maximum hops to 16 can cause a large number of nodes to be discovered rapidly, causing both short-term and long-term system performance problems. Unless you are sure your system is appropriately configured to handle the number of IPX nodes in your environment, Hewlett-Packard recommends that you expand your management domain slowly.

There is no way to add the router manually if it does not respond to IPX diagnostic requests.

3. If you increased your IPX hop count, but no nodes are being discovered in some or all of the IPX networks, check the following:
  - If the IPX networks exist on your map, verify that the target networks are managed. Use `Edit:Manage Objects` to manage the target networks.
  - If you have IPX discovery configured to run only at a certain time of day, new node discovery will not happen on the newly discovered networks until that time. You can verify your IPX discovery configuration using `Options: Network Polling Configuration:IP/IPX`

### Problems with Specific IPX Devices

**Some IPX nodes are missing from my map** If NNM discovers some, but not all nodes that support IPX:

1. Verify the installation and configuration of the IPX transport on the nodes that are not responding. In particular, check the IPX Transport Frame Type, making sure that it is either set to `Auto Detected` or set to match the Frame Type of the management system and servers.
2. Systems must respond to IPX diagnostic requests in order for NNM to add them to the databases and map. The following systems are known not to respond to IPX diagnostic requests:
  - Some dedicated routers (that is, not NetWare servers) do not support IPX diagnostic protocols.
  - Windows systems, including the management station itself.

To verify that a system responds to IPX diagnostic packets, you must know its IPX address, which is a combination of the IPX network number and its node hardware address. Once you know the IPX address, you can test connectivity to that IPX address by issuing the **`install_dir\bin\ipxping <IPX_address>`** command. For example: `install_dir\bin\ipxping 00000100:080009ABCDEF` sends an IPX diagnostic packet to the node with a LAN hardware address of 080009ABCDEF in the IPX network 00000100.

**IPX nodes have incorrect names or labels** NNM uses the following rules to choose the label for an IPX node:

- If the node has an IP hostname, then truncate the IP hostname to just the basename and use that—preference is given to the IP name of an object.
- If the node is a NetWare server, use the NetWare server name as the label. Otherwise, take the network number of the internal server address (that is, remove the 00000000000, which is the same for every server).
- If the node supports SNMP and reports an SNMP `sysName` value, use that as the label.
- If the node supports IP, use the IP address.
- If the node supports IPX, use the host-address portion of the IPX address, formatted to translate the vendor of the hardware. (For example, 100:080009ABCDEF gets a label of HP-ABCDEF).
- Otherwise, if the node has an LLA/MAC, use the hardware address as the label, formatting the hardware vendor if possible.

Based upon this information, troubleshoot naming problems by doing the following:

- If the node label is an IP address instead of a node name, update the IP address-to-hostname mapping. Verify the DNS configuration under the Network TCP/IP Configuration, and make sure your hostname resolution is configured correctly.
- If the node supports both IP and IPX and the label is incorrect, verify the IP address-to-hostname mapping as above.
- If the node is a Netware Server, verify that the correct server name is being propagated by the IPX network protocols by checking IPX SAP packets that show the server name, server type, and server address. You can check IPX SAP packets by doing either:
  - Listen for the information. Every server on the local network should broadcast a set of SAP packets once per minute. You can see the contents of these packets by typing the following at the command prompt:  
**`install_dir\bin\ipxping -s -l`**
  - Query for the information. You can obtain the information immediately by typing the following at the command prompt:  
**`install_dir\bin\ipxping -sbgv`**

If the name returned via either of the above methods does not match your expectations, you will need to reconfigure the IPX Server name.

- If the node supports SNMP, verify that the SNMP `sysName` is configured correctly on the target node. View the SNMP `sysName`, as determined by `netmon`, using `Edit: Object Properties`. Select the IP Map application and click `Edit Attributes`. The System Name attribute will be the SNMP `sysName` variable. If this name is blank, either the system does not have a `sysName` configured, or the SNMP configuration for the node is incorrect. Many nodes are shipped with a pre-configured SNMP `sysName` which may cause several nodes to have the same label.
- If the discovered label is a hardware address, but no vendor translation is occurring, check the contents of the following file:  
`install_dir\conf\physAddr.conf`

Make sure the vendor is listed correctly; for example: 080009 HP

**Nodes supporting both IP and IPX appear on the map as two separate nodes** If nodes that support both IP and IPX are showing up as two separate nodes on the map, verify your SNMP configuration for the IPX addresses of the node. The nodes can only be determined to be the same node if the node supports SNMP over IPX, and reports the MIB-II `ipAddrTable` correctly.

SNMP could be misconfigured over IPX. At the command prompt, type:  
**`snmpwalk IPXaddress system`**

If you do not receive a response, verify the community name configuration (see “GET- and SET-Community Name and SNMP Port Issues” on page 128).

**Duplicate IP Address messages are in the Configuration Alarms list** Some IPX nodes are shipped with the IP address 192.0.0.192 preconfigured. This normally does not cause problems for IP discovery, but might for IPX discovery. If multiple nodes support SNMP over IPX, and report the 192.0.0.192 address as a valid address, Duplicate IP Address alarms will be generated.

To fix this problem, configure a correct IP address on each IPX node in question.

To configure NNM to simply ignore the problem and prevent these particular Duplicate IP Address alarms from showing up in your Alarm Browser, do the following:

1. From any submap, select `Options:Event Configuration`. Click the `OpenView` entry in the `Enterprises` section.
2. Select `OV_Duplicate_IP_Addr` from the event list.
3. From the `Event Configuration` window, use `Edit:Events->Copy` to make a copy of the event. A window will be displayed with the event information to be modified.
4. On the `Description` tab, enter a new event name; for example, **Ignored\_Dup\_Addrs**.
5. Click `Only specified sources`.
6. Click the `Sources` tab. Identify each node that is generating duplicate IP address alarms by doing one or more of the following:
  - Select each offending IPX node on the submaps and click the `[Add From Map]` button in the event configuration dialog box.
  - Type each node name or address into the `Source` entry area, and click `Add`. The source name should be the name listed in the `Source` column of the `Alarm Browser`.
  - Create an ASCII file listing the full host name or IP addresses of each offending IPX source, one per line. (Use of wildcard characters is allowed in this file.)  
  
Type the full path and name of this file into the `Source` entry area, and click `Add`.File names and sources can be intermixed in this list. See NNM's online help for more information.
7. Click the `Event Message` tab.
  - a. Select either `Don't log or display`, or `Log only`.
  - b. Click `[OK]`.
  - c. Click `File:Save` to cause the event configuration to be updated.
  - d. Click `File:Exit` to exit the `Event Configuration` window.

## Stop Everything and Start Discovery Over Again

If a few aspects of your map do not look right, see “Troubleshooting Discovery” on page 114. However, if the entire map does not accurately represent your network or you decide to try discovering your network in a more limited fashion (page 107) and want to restart initial discovery, repopulate the databases, and generate a new map, do the following:

1. Stop all of NNM’s services (background processes):

---

### CAUTION

---

See *A Guide to Scalability and Distribution* for information about shutting down NNM in a distributed environment.

- *Windows:*  
Select Start:Programs:HP OpenView:  
Network Node Manager Admin->NNM Services - Stop.
  - *UNIX:* As root, at the command prompt, type **ovstop**.
2. Since this step deletes NNM’s topology, object, and map databases and your maps, you may wish to back up your existing databases before proceeding (see “Backup/Restore to Protect Your Investment of Time” on page 149).

If you have customized NNM’s configuration (discovery configuration, SNMP configuration, event configuration, or data collection and threshold monitoring, etc.) none of these configurations are affected by this step.

Execute the following:

- *Windows:* Use the Windows Explorer program to highlight the **install\_dir\databases\openview\** directory. Select all of the contents of that directory and delete the contents.
- *UNIX:* Type

```
cd $OV_DB/openview
rm -rf $OV_DB/openview/**
```



---

**NOTE**

---

If you are using a database other than the one embedded in NNM, see NNM's online manual, *Reporting and Data Analysis*.

3. Optional: To remove the old error messages and start again, execute the following:

- *Windows*: Use the Windows Explorer program to highlight the ***install\_dir\DATABASES\eventdb\*** directory. Select all of the contents of that directory and delete the contents.

- *UNIX*: Type

```
cd $OV_DB/eventdb
rm -rf $OV_DB/eventdb/*/*
```

---

**NOTE**

---

If you are using a database other than the one embedded in NNM, see NNM's online manual, *Reporting and Data Analysis*.

4. Restart NNM's services. NNM runs initial discovery, populates the databases, and generates a new map:

- *Windows*:  
Select Start:Programs:HP OpenView:  
Network Node Manager Admin->NNM Services - Start.
- *UNIX*: At the command prompt, type:

```
xnmsnmpconf -clearCache
$OV_BIN/ovstart ovwdb
$OV_BIN/ovw -fields
$OV_BIN/ovstart
```

See “Discover the Network... Let NNM Do It” on page 86 to start again.

Initial Network Discovery: Options and Troubleshooting  
**Stop Everything and Start Discovery Over Again**

---

## **6** **Preserve Your Sanity: Backup and Polling Configuration**

Before you begin to configure NNM to meet your team's needs, it is a good idea to make sure that the backup process for NNM is in place and working properly. For information about backing up NNM see page 149.

NNM monitors the devices on your network through standard network protocols. This means that network traffic is generated while accomplishing network management. NNM ships with a generic polling configuration already in place. Take time to learn about and customize the polling configuration to meet your particular business needs. For information about controlling the polling traffic generated by NNM see page 163.

## Backup/Restore to Protect Your Investment of Time

Backing up NNM can be integrated into your regularly scheduled network backup scheme. The backup happens in the background while your team continues to monitor alarms on your network. NNM temporarily pauses all services (background processes) whose activities result in changes to the NNM databases (such as the `ovw`, `ovwdb` and `ovtopmd` databases), thus ensuring consistency among the NNM databases and minimizing the potential for database corruption.

After the synchronized files are copied to the backup directory, the services resume normal activity. Then the files that do not require services to be in a paused state are copied to the backup directory.

During the backup of synchronized information, your team's maps are frozen in time, but the Alarm Browser's list and all Data Collection & Threshold information stay current throughout the backup.

---

### NOTE

Do not initiate changes to NNM's configuration files from the command line during a backup. It may cause data corruption in the backup copies of the files.

---

**You must ensure that the information copied to the backup directory is archived to the disaster-recovery media of your choice (such as tape).**

Add NNM's backup command to your list of regularly scheduled maintenance jobs. This script needs to run prior to your regularly scheduled backup process:

- *Windows:* `install_dir\bin\ovbackup.ovpl`
- *UNIX:* `$OV_BIN/ovbackup.ovpl`

Add NNM's backup directory to the list of items in your regularly-scheduled backup procedure. The backup directory (staging area) on the management station is:

- *Windows:* `install_dir\tmp\ovbackup\`

- *UNIX*: `$OV_TMP/ovbackup/`

See the *ovbackup.ovpl* and/or *ovrestore.ovpl* reference pages in NNM's online help (or the UNIX manpages) for more information.

## How Does the Backup Work?

The *ovbackup.ovpl* command encapsulates all the steps needed to create a backup copy of NNM's critical files. In addition, other HP OpenView programs may supply scripts to include their critical files in the backup process.

A backup consists of two phases. The purpose of these phases is to copy the directory structure and data to the backup staging area. Two distinct phases are required in order to ensure that NNM is paused for the shortest possible time.

- **Operational Phase**

NNM runs each script located in the *pre\_pause* directory. Affected applications take preparatory steps. Then, NNM broadcasts an *ovpause* command. *All* services configured to respond to NNM's *ovpause* command take appropriate action. In some cases, the service temporarily becomes inactive. In other cases, such as data collection (*snmpCollect*), the service is not completely inactive; data continues to be recorded in temporary files until the backup is complete.

Once all services registered to respond to the pause have successfully paused, NNM runs each script located in the *operational* directory. Because all affected services are paused, the data copied during this phase is *synchronized*.

Once all files are copied to the backup directory structure, NNM broadcasts the *ovresume* command to restart all services. Some services, such as the data collector (*snmpCollect*), then copy their data collected during the paused state from their temporary files to the regular files. In this way, even during a paused state, data collection is not interrupted. Finally, NNM runs each script located in the *post\_resume* directory to return affected applications to an operational state.

- **Analytical Phase**

NNM runs each script located in the `analytical` directory. An affected service may or may not be paused and resumed during the backup process, depending on the settings in its local registration file. The backup scripts might also pause and resume affected applications, themselves.

**It is your responsibility to ensure that your disaster recovery plan includes archiving NNM's backup directory during your regularly scheduled backup scheme.**

---

**CAUTION**

If you choose to run `ovbackup.ovpl`, you *must* deactivate the default `solid.ini` scheduled backup. Do not do both or neither. Running both programs will result in backups that may not restore properly. Running neither program may cause severe performance problems with the embedded database. See “Problem: Backup failed. SOLID Database Error 10019:Backup is already active” on page 159 or the *Reporting and Data Analysis with HP OpenView Network Node Manager* online manual for more information.

---

## Backing Up and Archiving All Critical Files

Try the backup and archive procedure now, before you spend a lot of time customizing NNM! Then back up and archive NNM files periodically while you configure NNM for your specific site, and back up and archive NNM files regularly as part of your disaster recovery plan.

1. The scripts named `nnm_checkpoint.ovpl` back up all of NNM's critical data. Additional scripts could be provided by other HP OpenView application developers or by you. Backup scripts must be placed in one of the following directories:

- *Windows:*

```
install_dir\conf\ovbackup\pre_pause or  
install_dir\conf\ovbackup\checkpoint\operational or  
install_dir\conf\ovbackup\checkpoint\analytical or  
install_dir\conf\ovbackup\post_resume
```

- *UNIX:*

```
$OV_CONF/ovbackup/pre_pause or  
$OV_CONF/ovbackup/checkpoint/operational or  
$OV_CONF/ovbackup/checkpoint/analytical or  
$OV_CONF/ovbackup/post_resume
```

---

**TIP**

To prevent NNM from being paused for longer than necessary, scripts should be placed in the *analytical* directory unless the data needs to be synchronized with the NNM databases. Scripts placed in the *operational* directory are run while all services registered to respond to the *ovpause* command are paused. Scripts placed in the *analytical* directory pause and resume services one at a time or in groups, only if necessary. See “Custom Scripts” on page 161 for information about writing your own backup/restore scripts.

- 
2. Check on disk space availability before backing up the databases.

At a minimum, the destination directory must have space for the NNM data that *ovbackup.ovpl* copies. If additional scripts have been added by other applications or the administrator, then additional space is needed. To determine the minimum space requirements, calculate the sum of the space used by NNM in the directories listed in Table 6-1 on page 153.

The *ovbackup.ovpl* script copies information to the following default directory:

- *Windows:* *install\_dir\tmp*
- *UNIX:* *\$OV\_TMP*

You can change the default directory by using the *ovbackup.ovpl -d* option. See the *ovbackup.ovpl* reference page in NNM’s online help (or the UNIX manpage) for more information.

---

**TIP**

As soon as you install NNM, data collection begins on certain MIB objects. Data collection consumes disk space. Backing up data collection files consumes additional disk space. Disk space being



consumed by any SNMP data collection as well as the back-up of these files needs to be managed. For more information on network performance data collection, see page 429.

**Table 6-1 Directories Included in Backup**

Windows	UNIX
<p><i>Operational files:</i></p> <p><code>install_dir\databases\openview\*.*</code>  <code>install_dir\databases\eventdb\*.*</code>  <code>install_dir\databases\nnmet\*</code>  <code>install_dir\log\*.*</code>  <code>install_dir\conf\*.*</code>  <code>install_dir\registration\*.*</code>  <code>install_dir\lrf\*.*</code>  <code>install_dir\fields\*.*</code>  <code>install_dir\symbols\*.*</code></p>	<p><i>Operational files:</i></p> <p><code>\$OV_DB/openview/*</code>  <code>\$OV_DB/eventdb/*</code>  <code>\$OV_DB/nnmet/*</code>  <code>\$OV_LOG/*</code>  <code>\$OV_CONF/*</code>  <code>\$OV_REGISTRATION/*</code>  <code>\$OV_LRF/*</code>  <code>\$OV_FIELDS/*</code>  <code>\$OV_SYMBOLS/*</code></p>
<p><i>Analytical files:</i></p> <p><code>install_dir\databases\snmpCollect\*.*</code>  <code>install_dir\databases\analysis\*.*</code></p>	<p><i>Analytical files:</i></p> <p><code>\$OV_DB/snmpCollect/*</code>  <code>\$OV_DB/analysis/*</code></p>

After the backup, the `ovbackup` directory contains a duplicate copy of all directory structures as well as individual files. The backup process does not compress data, so depending on the size of the original database, the directory could become very large:

- *Windows:* `install_dir\tmp\ovbackup\`
- *UNIX:* `$OV_TMP/ovbackup/`

You can specify a different directory with the `ovbackup.ovpl -d` option. See the `ovbackup.ovpl` reference page in NNM's online help (or the UNIX manpage) for more information.

**TIP**

The following *analytical data* directories are *not* included in NNM's backup scripts:

- Windows:

```
install_dir\backgrounds  
install_dir\bitmaps  
install_dir\www\htdocs\bitmaps  
install_dir\www\registration
```

- UNIX:

```
$OV_BACKGROUNDDS  
$OV_BITMAPS  
$OV_WWW/htdocs/bitmaps  
$OV_WWW_REG
```

If you make changes or additions to map backgrounds, bitmap files, or web registration files ensure that your new files are properly backed up. See “Custom Scripts” on page 161 for information about writing your own backup/restore scripts.

---

3. Decide how you will archive the backup copy of NNM’s critical files. Implement measures to ensure that the files in the `install_dir\tmp\ovbackup` directory (`$OV_TMP/ovbackup`) are archived to the disaster-recovery media of your choice on a regular and timely basis *and NOT during the time that the NNM backup scripts are running*.

---

#### CAUTION

*If you redirected the Data Warehouse storage from the SOLID database to another database of your choice:* The information will not be included by the `ovbackup.ovpl` script. Be sure to back up the database files that are used by NNM’s reporting feature and Data Warehouse feature. See your database provider’s documentation for directions about backing up your data.

---

4. At the command line, type `ovbackup.ovpl` or create a script that issues the `ovbackup.ovpl` command at the scheduled time of your choice. Refer to the `ovbackup.ovpl` reference page in NNM’s online help (or the UNIX manpage) for more information.

---

**NOTE**

Do not initiate changes to NNM's configuration files from the command line during a backup. It may cause data corruption in the backup copies of the files.

---

---

**TIP**

*UNIX systems only issue:* NNM's `ovw`, `ovwdb`, and `ovtopmd` databases use sparse files. Sparse files are RDBM files which are stored to disk with the NULs stripped out. Sparse files usually have the extension `.PAG`. NNM's backup program makes sure that sparse database files are *not* expanded. Do not use `tar` because it expands sparse files.

---

5. After the backup completes:

a. Check the ASCII log file to ensure that NNM's critical files were copied successfully:

- *Windows:* `install_dir\tmp\ovbackup.log`
- *UNIX:* `OV_TMP/ovbackup.log`

b. At the command line, type **ovstatus** to verify that all NNM services were successfully restarted.

6. Copy the files to the disaster recovery media of your choice.

7. Verify that your archive process successfully ran to completion and that the files on your disaster-recovery media are valid.

8. Optional: to reclaim hard drive space on your management station, delete the archived files from `install_dir\tmp\ovbackup\*.*` (`OV_TMP/ovbackup/*`).

## Restoring All of NNM (procedures/options)

In the event of a disaster or equipment upgrade, you must restore NNM to the most recently backed up state:

1. Install NNM from the installation CD, if necessary, and install any patches.
2. Stop all of NNM's services (background processes):

- *Windows:*  
Select Start:Programs:HP OpenView:  
Network Node Manager Admin->NNM Services - Stop.
  - *UNIX:* As root, at the command prompt, type **ovstop**.
3. Retrieve the latest NNM backup files from your disaster-recovery media and copy them to the management station's backup directory:
- *Windows:* `install_dir\tmp\ovbackup\`
  - *UNIX:* `$OV_TMP/ovbackup/`

---

**TIP**

*UNIX systems only issue:* NNM's `ovw`, `ovwdb`, and `ovtopmd` databases use sparse files. Sparse files are RDBM files which are stored to disk with the NULs stripped out. Sparse files usually have the extension `.PAG`. Make sure that the software being used to copy the files from your disaster-recovery media to the NNM management station does not expand sparse database files. Do not use `tar` because it expands sparse files.

---

4. Type **ovrestore.ovpl** from the command line to run the script files that update NNM's databases, log files, and configuration information files.

The scripts named `nnm_restore.ovpl` restore all files found in the `ovbackup` directory. Additional scripts could be provided by other OpenView application developers or by you. See "Custom Scripts" on page 161 for information about writing your own backup/restore scripts. These restore scripts must be placed in one of the following directories:

- *Windows:*  
`install_dir\conf\ovbackup\restore\operational`  
`install_dir\conf\ovbackup\restore\analytical`
- *UNIX:*  
`$OV_CONF/ovbackup/restore/operational`  
`$OV_CONF/ovbackup/restore/analytical`

5. Verify that your files are safely restored.

Optional: to reclaim hard drive space on your management station, delete the archived files from `install_dir\tmp\ovbackup\*.*` (`$OV_TMP/ovbackup/*`).

6. Start all of NNM's services (background processes):

- *Windows:*  
Select Start:Programs:HP OpenView:  
Network Node Manager Admin->NNM Services - Start.
- *UNIX:* As root, at the command prompt, type **ovstart**.

## Restoring Part of NNM (procedures/options)

The `ovrestore.ovpl` command allows you to specify that only the operational data (that must remain in a synchronized state) or only the analytical data be restored. It is not recommended that you restore smaller segments of NNM than these two categories. In fact, doing so could introduce data corruption.

See the `ovrestore.ovpl` reference page in NNM's online help (or the UNIX manpage) for more information. Remember that the following steps apply:

1. Stop all of NNM's services (background processes):

- *Windows:*  
Select Start:Programs:HP OpenView:  
Network Node Manager Admin->NNM Services - Stop.
- *UNIX:* As root, at the command prompt, type **ovstop**.

2. Retrieve the latest NNM backup information from your disaster-recovery media and copy it to the management station's backup directory:

- *Windows:* `install_dir\tmp\ovbackup\`
- *UNIX:* `$OV_TMP/ovbackup/`

---

### TIP

*UNIX systems only issue:* NNM's `ovw`, `ovwdb`, and `ovtopmd` databases use sparse files. Sparse files are RDBM files which are stored to disk with the NULs stripped out. Sparse files usually have the extension `.PAG`. Make sure that the software being used to copy the files from

your disaster-recovery media to the NNM management station does not expand sparse database files. Do not use `tar` because it expands sparse files.

- 
3. Type **ovrestore.ovpl** (along with the parameter specifying either operational or analytical data) from the command line to run the script files that replace NNM's databases, log files, and configuration information files with the backup copy.

The scripts named `nnm_restore.ovpl` restore NNM-specific data in either the operational or analytical files found in the `ovbackup` directory. Additional scripts could be provided by other HP OpenView application developers or by you. See "Custom Scripts" on page 161 for information about writing your own backup/restore scripts. These restore scripts must be placed in one of the following directories:

- *Windows:*

```
install_dir\conf\ovbackup\restore\operational  
install_dir\conf\ovbackup\restore\analytical
```

- *UNIX:*

```
$OV_CONF/ovbackup/restore/operational  
$OV_CONF/ovbackup/restore/analytical
```

4. Verify that your files are safely restored.

Optional: to reclaim hard drive space on your management station, delete the archived files from `install_dir\tmp\ovbackup\*.*` (`$OV_TMP/ovbackup/*`).

5. Start all of NNM's services (background processes):

- *Windows:*

```
Select Start:Programs:HP OpenView:  
Network Node Manager Admin->NNM Services - Start.
```

- *UNIX:* As root, at the command prompt, type **ovstart**.

## Troubleshooting Information

At the completion of the backup, ensure that the backup completed without the errors. This section describes some errors which can occur. Look at the last entry in NNM's backup log file:

- *Windows:* `install_dir\tmp\ovbackup.log`
- *UNIX:* `$OV_TMP/ovbackup.log`

**Problem:** The system is already in a paused state.

**Solution:** An error occurred while pausing the services. `ovpause` logs an error and generates a trap. The backup is aborted. From the command line, type `ovresume.ovpl` and then try the backup again. If you are experiencing time-out errors for a particular service during the pause process, double the length of its pause time-out by altering the affected service's `lrf` file. See the *lrf* reference page in NNM's online help (or the UNIX manpage) for more information. You must use the `ovaddobj` command to update NNM's configuration, and then use `ovstop` and `ovstart` to force NNM to acknowledge the change.

**Problem:** The backup failed due to a file copy error.

**Solution:** The original databases are not disturbed; however, the backup files cannot be used for a recovery. Fix the copy failure condition and then restart the backup. If the copy failed due to insufficient space, free up space in the staging area or use the `ovbackup.ovpl -d` option to copy files to a different directory. See the *ovbackup.ovpl* reference page in NNM's online help (or the UNIX manpage) for more information.

**Problem:** The system is not in a paused state.  
`ovresume` cannot continue.

**Solution:** The backup script assumes that there will be no operator interaction during the backup. Should someone on your team execute `ovresume` from the command line prior to the backup completing, the backup process may not run to completion. The original databases are not disturbed; however, the backup files cannot be used for a recovery. Restart the backup.

**Problem:** Resume attempt FAILED.

**Solution:** Check the backup log file. If you also see "Resume time-out" errors, see the *ovresume* reference page in NNM's online help (or the UNIX manpage) for information about the `-t` parameter to extend the time-out interval. If you see "Resume Error" messages, call HP customer support.

**Problem:** Backup failed. SOLID Database Error 10019:Backup is already active

**Solution:** If you are using NNM's reporting feature or Data Warehouse feature, historical information about your network is being stored in the SOLID database. By default, the SOLID database conducts a discreet backup of its own at 11 p.m. nightly. If you see this error in the `ovbackup.log` file, it means that the `ovbackup` was scheduled to run shortly after the SOLID database backup began.

To correct this problem you must deactivate the default `solid.ini` scheduled backup. The process for doing this is:

1. Copy the `$OV_DB/analysis/default/solid.ini` file to `$OV_DB/analysis/default/solid.ini.old` file.
2. Edit the `$OV_DB/analysis/default/solid.ini` file.
3. Comment out the `At=<time> backup` entry, by inserting a semicolon (;) at the beginning of the line. An example would be:

```
;At=23:00 backup
```

4. Save the `$OV_DB/analysis/default/solid.ini` file.

The embedded database will now be backed up only when the `ovbackup.ovpl` command is run. If the use of `ovbackup.ovpl` is stopped at a future time, the default backup should be reinstated by copying the `$OV_DB/analysis/default/solid.ini.old` back to `$OV_DB/analysis/default/solid.ini`.

**Windows-only Problem:** WARNING: Could not do 'secure' copy for file <filename>.

**Solution:** This is unavoidable when open files on the Windows operating system's NTFS are copied to the staging area. See the WARNINGS section of the `ovbackup.ovpl` reference page in NNM's online help for more information.

**UNIX-only Problem:** WARNING: Symbolic link errors or warnings from <directory>.

**Solution:** In most cases, this is not a problem and nothing needs to be done. It usually occurs when restore attempts to create a symbolic link for a file that is already there. See the WARNINGS section of the `ovbackup.ovpl(1m)` manpage for more information.



## Custom Scripts

NNM's backup procedure and restore procedure are designed so that they can be extended by placing any executable script into the appropriate operational or analytical directory. The HP-supplied `nmn_checkpoint.ovpl` and `nmn_restore.ovpl` scripts all have the extension `*.ovpl`. The suffix means "OpenView Perl" and refers to a specific subset of the Perl language used to implement NNM automated backup. This subset is not supported for general use by HP customers.

You may use the scripts supplied by HP as a reference or example while creating your own Perl scripts. However, if you create your own Perl scripts, do not use the `*.ovpl` extension in your script's file name and do not invoke HP's unsupported subset of Perl. Instead, ensure that you have a complete version of Perl installed on your system and use that one.

If you have *specific files* that you wish to include in NNM's backup process, the simplest way is to:

- **Windows:** Write a `.BAT` file that copies your files to the backup staging area, `install_dir\tmp\ovbackup`. Make sure that you duplicate the directory structure to your files, if necessary.
- **UNIX:** Write a shell script that copies your files to the backup staging area, `$OV_TMP/ovbackup`. Make sure that you duplicate the directory structure to your files, if necessary.

After you place your script in the appropriate directory, NNM runs it during the next backup or restore process:

- **Windows:**

```
install_dir\conf\ovbackup\pre_pause or  
install_dir\conf\ovbackup\checkpoint\operational or  
install_dir\conf\ovbackup\checkpoint\analytical or  
install_dir\conf\ovbackup\post_resume or  
install_dir\conf\ovbackup\restore\operational or  
install_dir\conf\ovbackup\restore\analytical
```

- **UNIX:**

```
$OV_CONF/ovbackup/pre_pause or  
$OV_CONF/ovbackup/checkpoint/operational or  
$OV_CONF/ovbackup/checkpoint/analytical or  
$OV_CONF/ovbackup/post_resume or  
$OV_CONF/ovbackup/restore/operational or  
$OV_CONF/ovbackup/restore/analytical
```

---

**CAUTION**

If you write scripts to copy *specific files* to the backup staging area, you must either:

- Create scripts that restore those specific files to the correct location (upon execution of `ovrestore.ovpl`). Place the scripts in the appropriate restore directories.
- Remember to copy the files to their original locations after running `ovrestore.ovpl`.

---

Refer to the *ovbackup.ovpl* or *ovrestore.ovpl* reference page in NNM's online help (or the UNIX manpage) for more information.

## Controlling the Amount of Traffic Generated by NNM

---

### NOTE

When you configure performance reports using the NNM web Reporting interface, the data collector starts collecting data. This data collection increases network traffic. You cannot control when or how frequently the collections are taken. If you need to reduce the amount of network traffic, you can suspend the generation of the report. See the NNM online help for more information.

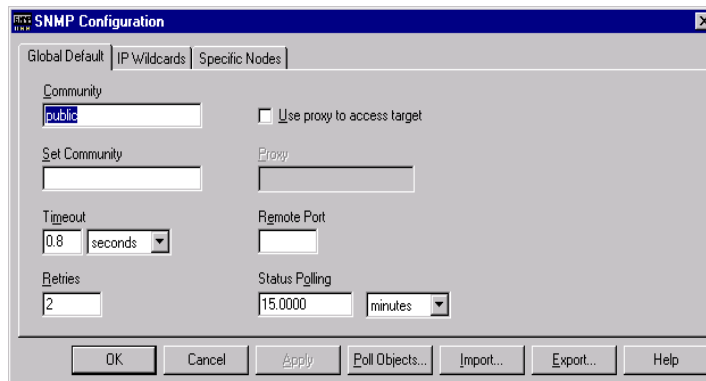
---

Network Node Manager software automatically generates intense polling traffic at first as the network monitoring process is discovering your network.

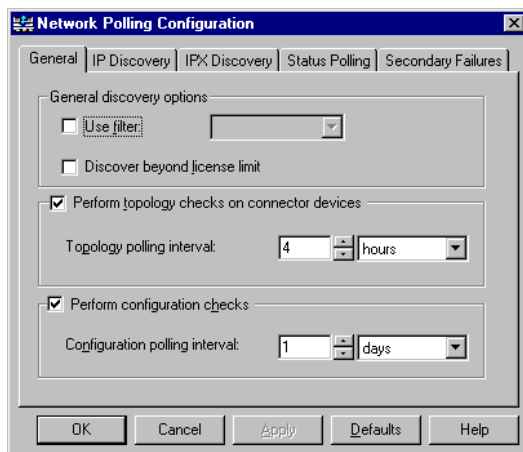
After NNM's initial discovery of your network, *you* are in control of NNM's polling based upon your business needs. There are five types of polling in NNM that you need to know about. You can customize polling, or you can turn polling off completely. There are benefits and consequences to your decisions: less polling means less network traffic; however, reducing the amount of polling traffic by lengthening polling intervals may delay real-time map updates, resulting in a less accurate map and less timely alarm logs. Turning polling off completely disables proactive management of your network.

You adjust NNM's polling through two dialog boxes accessed with the map menu's Options:SNMP Configuration and Options:Network Polling Configuration menu items.

**Figure 6-1** Options:SNMP Configuration Dialog Box



**Figure 6-2** Options:Network Polling Configuration Dialog Box



Check the online help from these dialog boxes for more detailed information than you will find in this chapter. The five types of polling are:

- **Status polling** (See “Status Polling” on page 166)

This is an ICMP ping sent to each interface of each managed device to verify if it is still accessible by the management station. In addition, a direct SNMP poll checks cached port tables on hubs and bridges to find the physical address listings of non-SNMP supported

devices, and a direct SNMP poll updates the designated list of DHCP addresses. Status polling intervals are configurable as described in this section. *Only management stations running a Windows operating system:* IPX diagnostic requests are also sent.

---

**TIP**

---

Level-2 devices (without their own IP address) whose physical addresses were found in cached port tables are included in NNM's databases and displayed on the map. However, no network polling traffic is generated on behalf of these devices.

- **Configuration check polling** (page 168)

This is an SNMP poll to gather current information about managed devices. This information is used by NNM to keep the maps current. The default interval is once-per-24-hours for each managed device.

- **Topology check polling of connecting devices** (page 169)

This is an SNMP poll to verify the manner in which managed connective devices (bridges and hubs via the Bridge MIB, Repeater MIB, and proprietary HP MIBs) service other devices on the network. The default interval is 4 hours.

- **New node discovery polling** (page 169)

There are two categories of new node discovery, IP and IPX. IP is an SNMP poll that scans for previously unknown devices and updates the map. The default interval is dynamic. *Only management stations running a Windows operating system:* if the IPX stack is configured as part of your Windows operating system, IPX issues RIP, SAP, and IPX diagnostic commands to detect additions to your management domain.

- **Secondary failures polling** (page 177)

This status polling works in conjunction with the Event-Correlation feature provided with NNM. During a network failure, NNM now identifies the device causing the problem and correlates failure messages from all other affected devices. Through this setting, you control how long NNM ignores devices other than the primary failure. You can also instruct NNM to never ignore certain critical devices, even if they are a secondary failure in an event stream.

- Also see suggestions for fine-tuning all of these polling settings on page 178.

## Status Polling

For each managed IP device, an ICMP ping is issued to verify if it is still accessible by the management station. The `Global Default interval` for IP devices is once every 15 minutes. Often, the `Global Default interval` setting for every device is not optimal. Refer to the list of mission-critical devices that you developed in Chapter 4, “Planning Your NNM Configuration.” You will probably want to poll mission-critical devices more frequently. Some customers decide not to poll certain devices at all, such as end-user PCs or certain printers.

Dynamic Host Configuration Protocol (DHCP) allows IP addresses to be allocated on a temporary basis (a lease). When the lease for an IP address expires, the address can be reused by a different node. This is useful in environments supporting mobile users who connect to the network with a laptop from many different places.

You can specify the range of IP addresses that your network is configured to assign dynamically for mobile devices. The default is `OFF` (no filter in use). If this filter is enabled, NNM keeps the map clean and the Alarm Browser list free of unnecessary messages about devices within this address range as they are repeatedly attached and detached from your network. Addresses are defined through a *filter* that you write using Boolean logic. See *A Guide to Scalability and Distribution* for information about writing, testing, and implementing the DHCP filter.

You can control status polling in the following ways.

### In the Options:SNMP Configuration dialog box

- Set polling intervals on specific nodes by IP address, IPX address, or hostname.

*For IP addresses only*, you can set polling intervals using IP address wildcards for a group of IP nodes (for example, `15.122.*.*`). You can also import a file that contains your configuration preferences; see the online help in this dialog box for information about using a file. The IP address wildcard is useful; for example, when you want to configure different values for time-outs or the number of retries for Wide Area Networks (WANs).

- Set the status polling interval to be used for any device not specifically listed. If you set it to 24-hours, polling begins at the time that you enter the setting. The actual time may drift depending on network conditions.
- Specify time-out and retry values. For example, you may want to increase the time-out and retry count values to prevent the management station from a premature time-out when making requests across a WAN because latency times are greater over WANs than LANs. Note that the time-out value is set in tenths of a second, and the time-out value doubles with each retry.
- Use filters to configure status polling by object classes. To view the current object class status polling interval configurations, click **Poll Objects** to open the **Status Polling Interval Configuration** dialog box. See the *OVfilterIntro* reference page (or the UNIX manpage) in NNM's online help for more information about defining filters. See the *netmon.statusIntervals* reference page (or the UNIX manpage) about modifying object-based status polling definitions.

---

**NOTE**

You may want to view or edit the current object class status polling interval configurations from a remote computer. To do this, use the following procedure:

1. Edit, as **root (UNIX)** or **Administrator (Windows)**, the following file and follow the instructions contained in the file.

- *Windows*: `install_dir\conf\remoteConfAllow.conf`
- *UNIX*: `$OV_CONF/remoteConfAllow.conf`

2. From your remote computer, point your web browser to the following location:

`http://hostname:3443/OvCgi/statusIntervalConf.ovpl`

---

**In the Options:Network Polling Configuration dialog box**

- Turn OFF/ON both IP and IPX status polling.
- Specify the interval of time that NNM should wait before deleting a “down node” from your maps. The default is 1 week.

- Identify your DHCP node list filter to specify a group of IP addresses that NNM should use for laptop computers. NNM handles this group of IP addresses differently, since devices will be connected and disconnected often from these addresses. Also specify the time interval that NNM should wait before deleting a disconnected node from your map.
- Designate your Discovery filter (if you created a filter of devices that should be excluded from polling). See “Create a Discovery Filter Identifying Which Devices to Include” on page 111.

---

**NOTE**

One other status polling method, **SNMP status polling**, is used only in special cases to obtain device status information using SNMP queries. SNMP status polling is not configured to run automatically at installation and does not generate any network traffic until you configure it to run. SNMP status polling is covered in more detail on page 259.

---

## **Configuration-Check Polling**

This type of polling gathers SNMP information about all currently managed devices for NNM’s exclusive use. Configuration-check polling gathers information such as:

- Contact and location change
- Forwarding IP packets change
- Interface added
- Interface deleted
- Incorrect routing by a node
- Link address change
- Mismatch of link address
- Network mask change
- Node name change
- Object Identifier change
- Mismatches in link addresses
- World wide web capabilities



This information is used by NNM to keep the maps and databases current. The default interval is once-per-24-hours. This setting applies to all managed devices; it cannot be customized per node.

You can control configuration-check polling in the following ways.

### **In the Options:Network Polling Configuration dialog box**

- Turn OFF/ON all configuration polling.
- Set the interval of time to wait between configuration polling, starting from the time that you set it. The time may drift depending on network conditions.

---

**TIP**

To monitor status of devices outside your firewall, see “SNMP Status Polling” on page 259.

---

## **Connector Topology Polling**

Connector topology polling monitors how hubs and bridges are connected in a network. All hubs and bridges are polled based upon the Bridge MIB (RFC1493), Repeater MIB (RFC2108), MAU MIB (RFC1515), and proprietary HP MIBs. You cannot set this polling on a per node basis. This setting helps keep your map current when you are moving devices around.

You can control connector topology polling in the following ways.

### **In the Options:Network Polling Configuration dialog box**

- Turn it OFF/ON.
- Specify the interval of time. The default is 4 hours.

## **New Node Discovery Polling (IP, Level-2, and IPX)**

New node discovery polling checks available ARP caches and Routing tables. This information is used to keep your maps current. This setting applies to all managed devices. It cannot be customized per node.

*If you are running NNM on a Windows management station:* For each managed IPX network, the following broadcasts occur for each discovery interval. The default interval is 6 hours:

- A single broadcast IPX Routing Information Protocol (RIP) request
- Two broadcast Service Advertising Protocol (SAP) requests
- Two broadcast IPX diagnostic requests per network, 10 seconds apart

You can control new node discovery polling in the following ways.

### **In the Options:Network Polling Configuration dialog box**

For IP and Level-2 devices:

- Turn new node discovery polling OFF/ON. The default is ON for both IP and Level-2 devices. See “Controlling Level-2 Device Discovery and Layout” for more information about configuring NNM to discover level-2 devices.
- Select either the auto-adjust discovery interval or set your own polling interval. The default is auto-adjust, which is based upon the number of new nodes discovered. The range is between 15 minutes and 24 hours.

For IPX devices:

- Turn new-node discovery polling OFF/ON. The default is OFF.
- Set your polling interval or the desired specific time of day for polling.
- Specify the maximum number of sequential routers through which to poll. The default hop count is zero.

## **Controlling Level-2 Device Discovery and Layout**

NNM discovers level 2 devices on both LANs and WANs and presents this information in a graphical format in the NNM map.

Table 6-2 on page 171 and Table 6-3 on page 175 summarize the information contained in this section. Table 6-2 shows the `netmon -k` keyword options and how these options work with the `Discover Level-2 Objects` check box to deliver the resulting NNM behaviors.

Table 6-3 shows how the `Discover New IP Nodes` check box and the `Discover Level-2 Objects` check box works with the `loadhosts` command to deliver the resulting NNM behaviors.

See “Controlling Level-2 Discovery: Detailed Information” on page 173 for more in-depth information about controlling level-2 discovery and layout.

**Table 6-2 Controlling Level 2 Discovery and Layout**

<b>netmon -k Option</b>	<b>Discover Level2 Object Check Box Status</b>	<b>NNM Behavior</b>
bridgeMIB=true	Checked	Level 2 nodes and interfaces are discovered and added to the topology database.  Bridge MIB layout is done.  Existing level 2 interfaces in the topology are unaffected.
bridgeMIB=false	Checked	Level 2 nodes and interfaces are discovered and added to the topology database.  No Bridge MIB layout is done.  Existing level 2 interfaces in the topology are unaffected.
bridgeMIB=true	Not checked	Level 2 interfaces are discovered and added to the topology database.  Level 2 nodes are not discovered.  Bridge MIB layout is done.  Existing level 2 interfaces in the topology are unaffected.
bridgeMIB=false	Not Checked	Level 2 nodes and interfaces are not discovered.  No Bridge MIB layout is done.  Existing level 2 interfaces in topology are unaffected.

**Table 6-2 Controlling Level 2 Discovery and Layout (Continued)**

<b>netmon -k Option</b>	<b>Discover Level2 Object Check Box Status</b>	<b>NNM Behavior</b>
discoverLevel2Nets=true	Checked	<p>L2Nets are created between pairs of routers with such routes.</p> <p>All level 2 interfaces on routers are discovered and added to the topology db.</p> <p>Existing level 2 interfaces in the topology are unaffected.</p>
discoverLevel2Nets=false	Checked	<p>L2Nets are not created between pairs of routers with such routes.</p> <p>All level 2 interfaces on routers are discovered and added to the topology db.</p> <p>Existing level 2 interfaces in the topology are unaffected.</p>
discoverLevel2Nets=true	Not checked	<p>Routers are examined for non-IP interfaces.</p> <p>L2Nets are created between pairs of routers with such routes.</p> <p>All level 2 interfaces on routers are discovered.</p> <p>Only level 2 interfaces on routers needed for L2Nets are added to the topology.</p> <p>Existing level 2 interfaces in the topology are unaffected.</p>

**Table 6-2 Controlling Level 2 Discovery and Layout (Continued)**

<b>netmon -k Option</b>	<b>Discover Level2 Object Check Box Status</b>	<b>NNM Behavior</b>
discoverLevel2Nets=false	Not Checked	Routers are not examined for non-IP interfaces. L2Nets are not created. No level 2 interfaces on routers are discovered. Existing level 2 interfaces in the topology are unaffected.

### **Controlling Level-2 Discovery: Detailed Information**

NNM controls the discovery and layout of level 2 devices in the following manner:

- NNM discovers and adds views of level 2 devices that support the Bridge MIB (RFC 1493).
- NNM examines routers for non-IP interfaces and adds views of level 2 networks that exist between pairs of routers to the NNM map.

You can control the discovery and map layout of level 2 devices by using of the following NNM controls:

- Using the `netmon -k` options.
- By selecting the `Discover Level2 Objects` check box in the IP Discovery area of the Network Polling Configuration user interface.

You can use the `Discover Level2 Objects` check box together with the `netmon -k bridgeMIB` options to configure the following discovery and layout behaviors:

- If the `bridgeMIB` keyword is set to true (the default value) and the `Discover Level-2 Objects` check box is checked, the `netmon` process discovers level 2 interfaces and adds these interfaces to the topology database. The `netmon` process attempts to create new

segments for bridges and switches supporting the bridge MIB and places the devices on the NNM map. In some cases the netmon process creates and places level 2 nodes on the NNM map. The netmon process changes the names of these newly created level 2 nodes from MAC layer addresses to IP or IPX name references if it determines that the level 2 nodes are IP or IPX nodes. Existing level 2 interfaces in the topology are unaffected.

- If you change the `bridgeMIB` keyword to `false` and the `Discover Level-2 Objects` check box is checked, the netmon process discovers level 2 interfaces and adds these interfaces to the topology database. The netmon process will not attempt to create new segments for the bridges and switches supporting the bridge MIB nor will it place the devices on the NNM map. Existing level 2 interfaces in the topology are unaffected.
- If the `bridgeMIB` keyword is set to `true` and the `Discover Level-2 Objects` check box is not checked, the netmon process discovers level 2 interfaces and adds these interfaces to the topology database. The netmon process attempts to create new segments for bridges and switches supporting the bridge MIB and places the devices on the NNM map. The netmon process will not create nor will it place level 2 nodes on the NNM map. Existing level 2 interfaces in the topology are unaffected.
- If you change the `bridgeMIB` keyword to `false` and the `Discover Level-2 Objects` check box is not checked, the netmon process will not discover level 2 interfaces nor will it add these interfaces to the topology database. The netmon process will not create nor will it place level 2 nodes on the NNM map. The netmon process will not attempt to create new segments for bridges and switches supporting the bridge MIB, nor will it place the devices on the NNM map. Existing level 2 interfaces in the topology are unaffected.

See the `ovtopofix` reference page in NNM's online help (or the UNIX manpage) for information on reversing the effects of the `bridgeMIB` keyword. Look for information on the `ovtopofix -l (el)` and `-I` options.

You can use the `Discover Level2 Objects` check box together with the `netmon -k discoverLevel2Nets` options to configure the following discovery and layout behaviors:

- If the `discoverLevel2Nets` keyword is set to `true` (the default value) and the `Discover Level-2 Objects` check box is checked, the netmon process examines routers for non-IP interfaces and creates

L2Nets between pairs of routers with non-IP interfaces. All level 2 interfaces on routers are discovered and added to the topology database.

- If you change the `discoverLevel2Nets` keyword to false and the `Discover Level-2 Objects` check box is checked, the `netmon` process will not examine routers for non-IP interfaces nor will it create L2Nets between pairs of routers with non-IP interfaces. All level 2 interfaces on routers are discovered and added to the topology database.
- If the `discoverLevel2Nets` keyword is set to true and the `Discover Level-2 Objects` check box is not checked, the `netmon` process examines routers for non-IP interfaces and creates L2Nets between pairs of routers with non-IP interfaces. All level 2 interfaces on routers are discovered. Only level 2 interfaces on routers needed for L2Nets are added to the topology. Existing level 2 interfaces in the topology are unaffected.
- If you change the `discoverLevel2Nets` keyword to false and the `Discover Level-2 Objects` check box is not checked, the `netmon` process will not examine routers for non-IP interfaces nor will it create L2Nets between pairs of routers with non-IP interfaces. No level 2 interfaces on routers are discovered. Existing level 2 interfaces in the topology are unaffected.

When the `Discover New IP Nodes` check box is not checked, the `Discover Level-2 Objects` check box is grayed-out, but still impacts NNM behavior. Table 6-3 shows the resulting NNM behavior when the `Discover New IP Nodes` check box is not checked and new nodes are added by using the `$OV_BIN/loadhosts` command, `netmon`'s seedfile, or when manually adding symbols to the map. The discovery of level 2 interfaces on these nodes is still controlled by the `Discover Level-2 Objects` check box.

**Table 6-3** Discovery Impact of Using `loadhosts` Command or a Seed File

<b>Discover Level-2 Objects Check Box Status</b>	<b>Discover New IP Nodes Check Box Status</b>	<b>NNM Behavior</b>
Checked	Not checked	Adds both IP interfaces and level 2 interfaces

**Table 6-3** Discovery Impact of Using `loadhosts` Command or a Seed File (Continued)

Discover Level-2 Objects Check Box Status	Discover New IP Nodes Check Box Status	NNM Behavior
Not checked	Not checked	Adds only IP interfaces

### Changing `netmon -k` Options

You can use the following procedure to modify the `netmon -k` options:

1. Log in as:
  - *Windows*: Administrator
  - *UNIX*: root
2. Open the `netmon` local registration file in the ASCII editing software of your choice:
  - *Windows*: `install_dir\lrf\netmon.lrf`
  - *UNIX*: `$OV_LRF/netmon.lrf`
3. Add the text shown in bold to set the keyword to the stated value:
  - To set `bridgeMIB= true`:

```
netmon:netmon:
OVs_YES_START:ovtopmd,pmd,ovwdb:-P -k segRedux=true -k
bridgeMIB=true:OVs_WELL_BEHAVED:15:PAUSE::
```

- To set `bridgeMIB=false`:

```
netmon:netmon:
OVs_YES_START:ovtopmd,pmd,ovwdb:-P -k segRedux=true -k
bridgeMIB=false:OVs_WELL_BEHAVED:15:PAUSE::
```

- To set `DiscoverL2Nets=true`:

```
netmon:netmon:
OVs_YES_START:ovtopmd,pmd,ovwdb:-P -k segRedux=true -k
discoverLevel2Nets=true:OVs_WELL_BEHAVED:15:PAUSE::
```

- To set `DiscoverL2Nets=false`;

```
netmon:netmon:
```



```
OVs_YES_START:ovtopmd,pmd,ovwdb:-P -k segRedux=true -k discoverLevel2Nets=false:  
OVs_WELL_BEHAVED:15:PAUSE: :
```

4. Turn off the netmon service (background process). At the command prompt, run **ovstop -c netmon**.
5. Force NNM to acknowledge the change. At the command prompt, run:
  - *Windows:* **ovaddobj install\_dir\lrf\netmon.lrf**
  - *UNIX:* **ovaddobj \$OV\_BIN/ovaddobj \$OV\_LRF/netmon.lrf**
6. Turn on the netmon service (background process): At the command prompt, run: **ovstart -c netmon**

See the *netmon* reference page in NNM's online help (or the UNIX manpage) for more information; look for the -k keyword parameter and the bridgeMIB and discoverLevel2Nets keyword arguments.

## Secondary Failure Polling

This setting allows you to fine-tune the behavior of the connector-down correlation circuit. The default is ON. See “NNM's Event Reduction Capabilities” on page 340 and “Connector Down Correlation” on page 350 for more information about the event correlation feature.

You can control secondary failure polling in the following ways.

### In the Options:Network Polling Configuration dialog box

- Turn secondary failure polling OFF/ON.
- Specify the factor by which the normal polling interval should be extended for secondary devices during a failure.
- Designate the section that you wrote in the *filters* file that specifies which critical devices should always be excluded from the secondary status list regardless of their location during a failure event.

See the *OVfilterIntro* reference page (or the UNIX manpage) in NNM's online help for more information about defining filters. See also *A Guide to Scalability and Distribution* for information about creating filters.

- Control the behavior of the map and event log during a failure event.

## Fine-Tuning the Polling Services

The amount of network management traffic generated by status polling depends on your network's configuration, including:

- How many nodes are up and how many are down?
- How many nodes are on the same subnet?
- How many nodes are on other subnets?
- How large is the permanent ARP cache?
- How probable is it that a particular IP address can be found in the permanent ARP cache?

Polling is controlled by NNM's `netmon` service (background process). Because automatic polling is based upon managed nodes, you can reduce the amount of network-management traffic by unmanaging nodes. Simply select the node or nodes on the map, and then select the `Edit:Unmanage Objects` menu item. You can add nodes and change nodes from unmanaged to managed. It may take a while for polling changes to take effect. Polling is scheduled on a per node basis, but the `netmon` service can fall behind schedule if its polling load exceeds capacity limits.

---

### TIP

To automatically set specific types of devices to unmanaged, such as all printers or all end-user systems, see the `oid_to_type` reference page in NNM's online help (or the UNIX manpage). See also "Modify the `oid_to_type` File to Unmanage Devices by Type" on page 112.

---

To determine if the polling process is overloaded, select the `Performance:Network Polling Statistics` menu item. If "Seconds until next status poll" or "Seconds until next SNMP poll" is a negative number, then the polling queue is overloaded and running behind.

If polling is continually running behind, you may want to change the startup parameter settings in the `netmon.lrf` file for queue length (`-q` controls the ICMP queue length, `-Q` controls the SNMP queue length, and `-x` controls the IPX queue length). The default queue setting is 3 on a management station running the Windows operating system and 20 on a management station running a UNIX operating system. Increasing queue length affects the management station's use of system resources and use of network bandwidth for network management.

For information about parameter choices available to you for configuring NNM's polling service:

- See the *netmon* reference page in NNM's online help (or the UNIX manpage) for a wealth of choices about controlling polling from the command line and start-up parameter settings. The information in this reference page lists the parameters allowed in the `netmon.lrf` file. See "Create a Seed File Specifying Multiple IP Networks to Manage" on page 103 for information about the process of changing the `netmon.lrf` file (the seed file requires a modification to the `netmon.lrf` file).
- See the *xnmpolling* reference page in NNM's online help (or the UNIX manpage) for information about how to change polling configuration from the command line.
- See NNM's online help available from within the Options:SNMP Configuration dialog box and the Options:Network Polling Configuration dialog box for more information.

Information that is gathered during polling is stored in NNM's `ovsnmp.conf_db` database.

## Watching NNM's Polling Queue

You can automate NNM to notify you whenever the `netmon` service is falling behind in the polling queue.

Use NNM's Data Collection & Thresholds feature (see page 429) to monitor two MIB values from HP's `netmon` MIB (provided on NNM's installation CD; use Load/Unload MIBs to access the `netmon` MIB):

- `nmICMPSecsUntilNextPoll`  
(.1.3.6.1.4.1.11.2.17.4.4.1.1.3) and
- `nmSNMPSecsUntilNextPoll`  
(.1.3.6.1.4.1.11.2.17.4.4.1.2.3)

Set the threshold configuration to generate a trap each time the polling queue falls behind to the negative amount that you specify.

Then use NNM's Event Configuration feature (page 404) to set up Automatic Action instructions for that trap instructing NNM to email a message to you or to page you each time that trap is received.

## Running NNM Without Network Polling

It is possible to run NNM without the `netmon` service (background process) that polls your network to generate the map and detect changes. This may be useful if the only applications you have running under NNM do not rely on `netmon`'s discovery functionality or if you are relying on collection stations to perform discovery and monitoring.

---

### NOTE

See *A Guide to Scalability and Distribution* for information about collection stations and how to proceed with this step when using collection stations.

---

To run NNM without automatic IP discovery, configuration checking, and polling by `netmon`, follow these steps:

1. Stop the currently running `netmon` service (background process).  
From the command line, type **`ovstop -v netmon`**
2. Remove `netmon` from NNM's start-up options. At the command prompt, type:  
*Windows:* **`ovdelobj install_dir\lrf\netmon.lrf`**  
*UNIX:* **`ovdelobj $OV_LRF/netmon.lrf`**

---

### NOTE

If you change your mind and want to restart `netmon` at a later time:

1. Add `netmon` to NNM's start-up options. At the command prompt, type:  
*Windows:* **`ovaddobj install_dir\lrf\netmon.lrf`**  
*UNIX:* **`ovaddobj $OV_LRF/netmon.lrf`**
2. Start the `netmon` service (background process):  
*Windows:* Select Start:Programs:HP OpenView:Network Node Manager Admin->NNM Services - Start.  
*UNIX:* As root, at the command prompt, type **`ovstart`**.

The `pmd`, `ovwdb`, and `ovtopmd` services must also be running before `netmon` can work properly. To generate a list showing the status of each of NNM's services:

- *Windows:* Select Start:Programs:HP OpenView:Network Node Manager Admin->NNM Services - Status.
- *UNIX:* At the command line prompt, type **ovstatus -c**.

If any of the processes are not successfully running, see Appendix A, “NNM Services and Files,” on page 523 for information about troubleshooting NNM’s services before continuing.

---



---

# **7** **Map Making Fundamentals**

The following concepts are useful to understand before you start creating your own customized maps:

- maps versus submaps (page 185)
- objects versus symbols (page 189)



## Maps versus Submaps

You can think of the relationship between maps and submaps as being much like the relationships between an atlas and its pages. The atlas is the map. The pages of the atlas are the submaps where you may view a particular continent, country, state, city, or even specific parts of a city. In NNM, when you view a part of your network map, you are actually viewing a submap. The view may be presented in a high-level submap that represents your entire network that spans the world, or in a more detailed submap of any portion of the network.

### Maps

A **map** is a set of related objects, symbols, and submaps that provides a graphical and hierarchical presentation of your network and its systems. You can create multiple maps, but only one map is open at a time for any given session of NNM.

You do not view a map directly; instead, you always view the submaps that comprise the map. You can display multiple submaps at any given time. Submaps are typically organized hierarchically to show an increasing level of detail.

Different maps can be used for defining different management regions, or for different presentations of the same management region. Different maps can be tailored to the needs of individual users.

You can create multiple maps and customize how information about objects is displayed on each map. Different maps can display information about the same object because maps obtain their information from the same source, the object database. In NNM, you can create new maps, delete maps, and choose the map to display from existing maps.

When you start NNM, a map is automatically opened. You can specify a map by name, or you can let the `ovw` service open a default map. While a map is open for display, it is simply called an “opened map.”

### Pros/Cons of Multiple Maps

You can view your various maps through multiple NNM SESSIONS. HOWEVER, Only one map can be opened in a single session at any given time. A single user can open multiple maps by opening multiple NNM sessions. Different users can open the same map (at the same time) through different NNM sessions.

When multiple users access the same map, all users except the first one to access the map are limited to Read-Only access. This means that only the first users will be able to create map snapshots or access any modification features that you, the map administrator, made available to them.

### Submaps

When you open a map, you actually view submaps of the map.

A **submap** is a particular view of the network environment. It consists of related symbols that are displayed in a single window. Each submap displays a different perspective of your map. NNM creates a root submap for each map. The root submap provides a standard, top-level submap for every map. Submaps are often organized in a hierarchical fashion for a given map, with the root submap at the top. You can also create independent submaps that are not associated with a hierarchy.

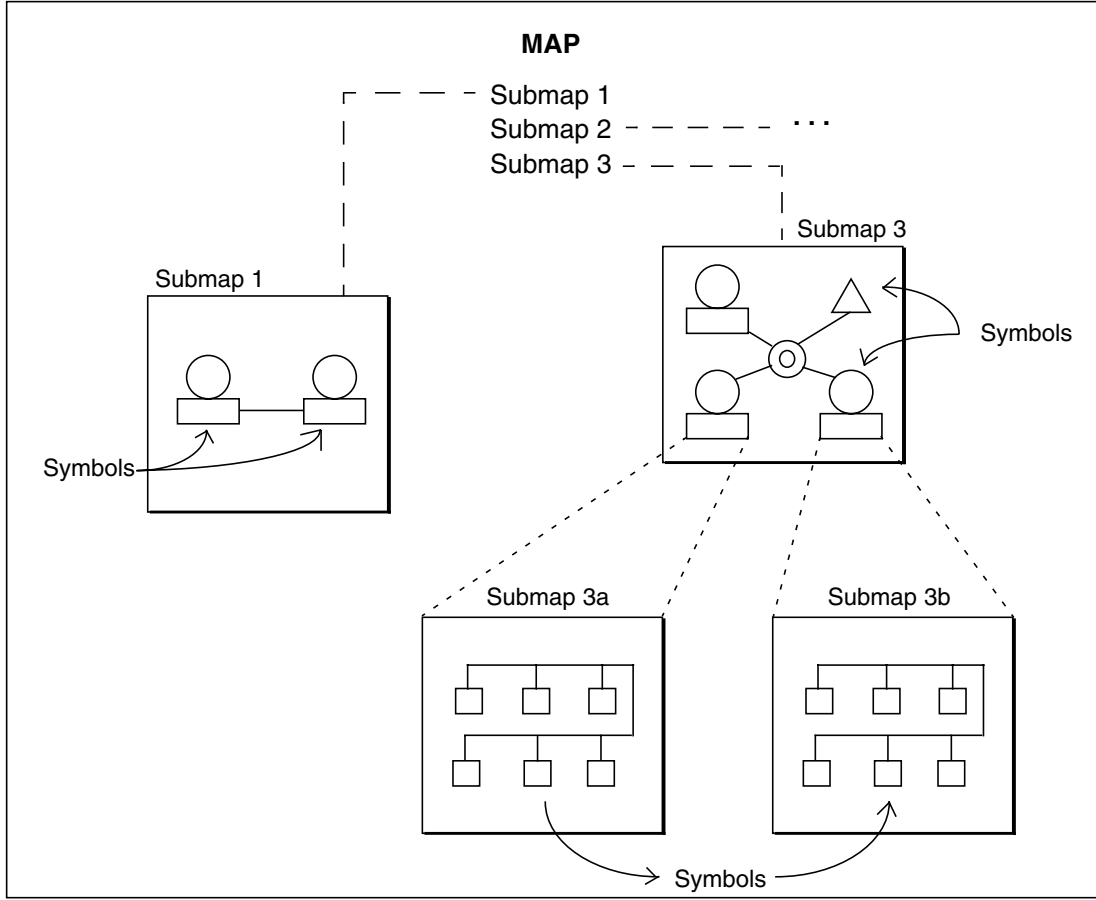
You can open and display multiple submaps of the open map at any given time, either by listing all the submaps of the open map and selecting the submaps to open, or by navigating from one submap to another. You can navigate among the submaps of the open map by double-clicking the mouse on explodable symbols. Double-clicking on an explodable symbol opens a submap that displays a more detailed view.

The hierarchical relationship of submaps creates a parent-child relationship between them. A submap may have several child submaps. The hierarchical relationship of submaps enables you to view your network from a distance, or to choose a more detailed view. For example, consider a submap that contains a single symbol that represents an entire organization. From this high-level view of the map, you can double-click on the symbol to open a child submap. The child submap may display a view of your network map from the perspective of a particular location. From there, you can select a specific department,

then a specific node. You can customize the organization of submaps in a map to suit your purposes; for example, to reflect the organization of your company.

The following figure shows how submaps and their symbols can be used to graphically display network or system management information in a network map.

**Figure 7-1 Symbols and Submaps in a Map**



In summary, a submap is a particular view of a map. Together, symbols, submaps, and maps make up the presentation of NNM.

### **Root Submap**

The root submap is the highest-level submap of the map. The first time a map is opened, the root submap is set as the default home submap. The root submap is a system-created submap that allows for the placement of very high-level objects by multiple applications. The root submap cannot be deleted. You can easily return to the root submap at any time by pressing the Home toolbar button.

### **Home Submap**

The home submap is the submap that appears first in a submap window when you open a map. It is analogous to a home directory. You can assign any submap of the map as the home submap.

### **Background Graphics**

A background graphic, such as a map or picture, may be displayed in the background plane on a submap window. The background graphic may be different for each submap. For example, you could use the floor plan of each floor of each building as the background graphic over which the placement of your network devices is shown.

## Understanding Objects versus Symbols

An **object** represents a particular entity or resource in a networked systems environment. An object might represent a physical piece of equipment on the network, the components of a node on the network, or parts of the network itself. The object represents the resource by modeling the characteristics (attributes) of the resource. An object exists on a map by virtue of being represented by a symbol on a submap in a map.

A **symbol** is a graphical representation of an object. A single object can be represented by multiple symbols. Multiple symbols for the same object can exist on the same submap, on multiple submaps of the same map, or on submaps of different maps. This enables multiple users on different maps to view a symbol of the same object at the same time. A symbol never represents more than one object at a time.

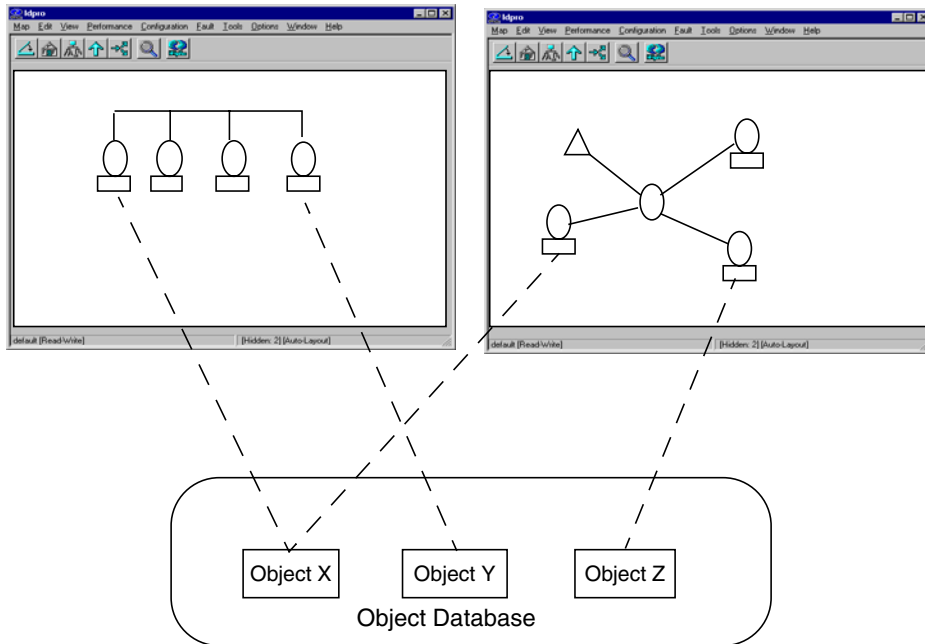
In addition to representing objects, symbols have other functions:

- Symbols let you navigate through the submaps of a map. Most symbols are explodable—when you double-click on an explodable symbol, a new submap window opens to let you “look inside” the object represented by the symbol.
- Some symbols execute actions. When you double-click on an executable symbol, a predefined action is executed on a predefined target.
- Symbols can be configured to reflect the status of the object that they represent or of objects in child submaps.

The following figure shows how symbols and objects are related. This figure contains two submaps that contain different symbols. Object Y is represented by a single symbol in the root submap, while Object X is

represented by symbols in both the root submap and submap2. Changes made to object X, such as a change in status, can be displayed in both symbols in the two submaps.

**Figure 7-2** Object-Symbol Relationships



## Objects

An object represents a logical or physical entity or resource, or a group of such logical or physical entities or resources that exist in a network environment. An object usually represents any particular thing of interest for the purpose of doing network or systems management. An object can represent a physical item (such as a PC, a workstation, a gateway, a router, an interface card, or an RS-232 connection), or it can represent a logical item (such as a group of PCs, all 486 PCs, or all nodes in a single department).

## Object Attributes/Properties

Each object stored in the map database contains attributes that define the object. An **attribute** is a characteristic of an object to which values can be assigned.

You can think of an attribute as a field that has a specific value. In an application dialog box that displays certain attributes of an object, the labeled fields represent the object's attributes, and the data in the fields are the object's attribute values, such as:

- hostname
- address
- status
- description
- owner

Every object has a special attribute called a **selection name**. The selection name is a textual name that uniquely identifies the object. An object can have multiple names, but each name must be unique for its name space. For example, an object might have a hostname (for TCP/IP networks) and a fully distinguished name (for OSI networks). Either name could be used as the selection name.

The phrase **selection list** appears throughout this manual and relates to objects. The selection list is a list of objects that are represented by the symbols selected by the user. The selection list is one of the primary ways that you pass arguments to NNM.

The attributes of an object are displayed via the `Edit:Object Properties` dialog box. This dialog box displays the selection name of the object and displays any comments that you, or another user, may have entered about this object.

In the `Edit:Object Properties` dialog box, there is a list of attribute categories, such as `General` and `Capabilities`. These categories are accessed by applications that may be managing or monitoring certain attributes about the object. Applications may also add their own attribute category. You can view/modify the object attributes in each category.

- The `General Attributes` dialog box provides a place for applications and users to display attributes in one common location. The `General Attributes` dialog box may display attributes of an

object from multiple applications. These fields may also be displayed in application-specific dialog boxes. The specific fields that are displayed in this dialog box depend on the applications that register a field for this dialog box. Check NNM's online help in this dialog box for more information.

You can set the values of these attributes from the dialog box. Note that values set in this dialog box are not verified. Applications or users may add additional fields to this dialog box by setting a general flag for the field in the field registration file.

- The Capabilities dialog box includes the object attributes that determine menu graying. The capabilities of the objects on the current selection list determine which menu items are enabled or disabled. All attributes displayed in the Capabilities dialog box are read-only.

The default for this dialog box is that unset capability fields are not displayed. You can change the behavior of this dialog box so that the unset fields *are* displayed by modifying your Windows system registry or UNIX \$APP\_DEFS/OVw file and changing the value of showUnsetCapabilitiesFields to TRUE.

The same information is available in the Edit:Add Object dialog box. This enables you to assign attribute values during object creation.

### Operations on Objects

Check NNM's online help for information about the operations you may commonly perform on objects, such as:

- adding an object
- selecting one or more objects
- locating objects
- adding attributes to an object
- changing an object's description
- changing the vendor and SNMP agent values
- deleting an object
- hiding an object
- showing a hidden object



## Symbols

A symbol provides a graphical representation of a particular object. NNM uses two varieties of symbols for display on the map: icon symbols and connection symbols.

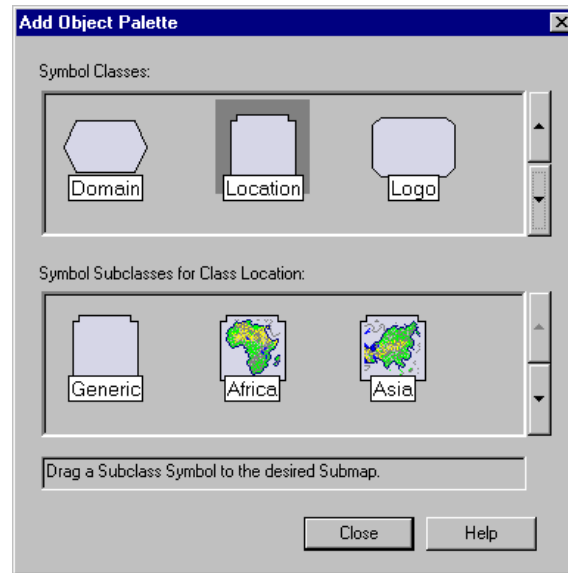
- *Icon symbols*, usually referred to simply as a symbol, consists of a geometric shape. A symbol graphic, or icon, usually appears inside the shape.
- *Connection symbols* connect two icon symbols or an icon symbol and a backbone. On a submap, connection symbols appear as lines. However, connection symbols are not merely lines. They represent objects and can display status. NNM uses connection symbols to show the current state of connections between resources on your network.

## Icon Symbols

Icon symbols include many symbol types. A symbol type consists of the symbol class and symbol subclass. The symbol class is indicated by the outside shape of the symbol. Each symbol class is further divided into

subclasses. Figure 7-3 displays the palette as it appears when you select the `Location` class. The subclasses of the selected symbol class are displayed in the lower portion of the palette.

**Figure 7-3** Subclasses in the Location Class



NNM contains a variety of predefined symbols. You can view the available symbols by browsing through the palette or by looking at the Display Legend dialog box. To view the legend, select `Help:Display Legend` from the menu bar. The classes and respective subclasses that are registered by NNM are listed in the Add Object Palette.

### Connection Symbols

A connection symbol is a line that graphically connects two icon symbols or an icon symbol and a backbone (a bus or a ring backbone) on a submap. The connection symbol displays the status of a connection between two objects. Connection symbols behave like icon symbols in the following ways:

- A connection symbol represents an object, just like an icon symbol.
- Connection symbols may reside on the application plane or the user plane.

- You can display a symbol pop-up menu for the connection, and open the *Symbol Properties* dialog box to view or change symbol characteristics.
- You can assign a label to the connection symbol.
- You can change the type of a connection symbol.

Like an icon symbol, the connection symbol may be explodable or executable.

### Symbol Characteristics

The following list discusses characteristics of symbols.

Symbol Variety	Symbols come in two varieties: the icon symbol and the connection symbol.	
Symbol Type	Symbol type consists of the symbol class and subclass.	
	Class	You can distinguish the symbol class by the outer shape of the symbol. For example, a circle, square, diamond, and so on.
	Subclass	A subclass is a further definition of the class. Each symbol class has a unique set of subclasses.
Status	NNM displays status information by changing the color of the class (or outer shape) of the symbol. The ten status states are categorized as either administrative or operational (see page 256).	
Label	You can label each symbol that appears on your network map. The label appears below the symbol. You can choose whether or not to display the label of a symbol.	
Location	A symbol can reside on one of two planes: the application plane or the user plane. If an object is not managed by any application, then symbols of that object reside on the user plane of the submap and have a box displayed around them. If an object is managed by one or more applications in a particular submap, then symbols of that object reside on the application	

plane of that submap. This enables you to visually distinguish between objects that are being managed by applications and those objects that are not.

Behavior	Behavior defines how the symbol behaves when you double-click on the symbol. A symbol may behave in one of two ways. It may open a child submap (explode the child submap), or it may execute an application.
Status Source	A symbol may present status information from one of three sources (see page 259). This characteristic enables applications to have more control over the presentation of status. In general, you do not need to change these settings.

### Operations on Symbols

Check NNM's online help for information about the operations you may commonly perform on symbols. Many of these symbol operations can be performed from the symbol pop-up menu:

- Adding a connection symbol
- Adding an executable symbol
- Adding an icon symbol
- Changing symbol behavior
- Changing a symbol label
- Changing a symbol type
- Cutting and pasting symbols
- Copying a symbol

See also Appendix C, "Changing All the Symbols for a Particular Device," on page 623.

---

## **8Map Customization**

## Putting It All Together

After you create the initial map and clean up any network problems that were identified, you have the *default* map. (See Chapter 5, Initial Network Discovery: Options and Troubleshooting, if you need to create the default map.) The purpose of the default map is to display a map of your network quickly when you first run NNM. Now you are ready to create copies of the default map and customize the copies for your specific needs.

You can create multiple maps and customize the way in which information about objects is displayed on each map. This enables you to distribute resources over your map to better match how your network and/or team is organized. Multiple maps can display information about the same object because all maps obtain their information from the same source, the object database.

### Which Maps for Which Users

In Chapter 4, Planning Your NNM Configuration, you identified the groups or individuals within your organization who need access to the maps. You also identified which aspects of your network each group or individual needs to see on their maps.

### Identifying the Purpose of Each Map

As you develop your custom maps, keep in mind the type of problem each group or individual is trying to prevent or solve by using the map. Design their map so that the amount of information displayed is limited to the minimum relevant information and that the information is presented in such a way that it makes their job easier. For example: the map for printer maintenance may show only printers and print servers with backgrounds showing actual floor plans to aid quick response efforts; the map for network troubleshooters might need to contain network infrastructure devices across the entire company; and the map for upper-management to use during client visits might need to show everything possible to look large and impressive.

## Determining How Many People Need Access to Each Map

In Chapter 4, you built a model of your customer base and network support team to understand how NNM can be customized to most effectively help your team.

Can any of the groups that you identified share a map, or do their information needs require completely different maps?

Keep in mind that a number of people can view one map simultaneously or multiple maps by using remote console access (see *A Guide to Scalability and Distribution* for information) or by using NNM's new web access (see Chapter 14, NNM on the Web,). When multiple people are accessing a single map, only the first user to open the map has read-write permission (provided that they have read-write permission to the map file), which means that only one person will be able to make changes to the map. However, everyone accessing NNM can acknowledge/delete alarms from the Alarm Browser as issues are resolved.

The number of remote access logins that can be supported is dependent upon a number of factors. See the *Performance and Configuration Guide* that was included with NNM's installation package.

## Planning Your Efforts

The remainder of this chapter presents a series of possible map customizing changes. Focus on customizations that help your team meet a business need. After initial map customizing, NNM users tend to use the Alarm Browser on a day-to-day basis and only use the map while troubleshooting a specific network alarm. So keep map customizations to a workable minimum. The customizing options are:

- “Copying the Default Map” on page 201
- “Controlling the Display of Devices Attached to Switches or Bridges” on page 203
- “Giving Your Network Symbols Meaningful Names” on page 207
- “Turning Connection Labels on or off” on page 209
- “Configuring Trunking and Meshing” on page 211
- “Establishing Submap Persistence Settings” on page 215
- “Controlling Which Devices Appear on the Map” on page 218

- “Changing/Adding Object Attribute Fields” on page 221
- “Making the Maps Look Like Your World” on page 226
- “Creating Your Own Map Symbols” on page 249
- “Specifying the Placement and Size of the Submap Windows” on page 252
- “Miscellaneous Configuration Changes” on page 254
- “Controlling Symbol Status” on page 256
- “Creating New NNM Features to Meet Your Team’s Needs” on page 264

Customizing the maps requires a substantial time commitment and should be done by someone who understands NNM and your network. To protect your efforts, you may want to remove functionality from the maps so that your team cannot accidentally make changes to your customizations. See Chapter 9, “Controlling Map Access,” on page 269.



## Copying the Default Map

When you start NNM, the default map is automatically opened with the home submap displayed. You can create multiple maps and specify which map NNM displays initially. Only one map can be opened in a single session at any given time. Different users can open the same map (at the same time) through different NNM sessions, although only the first session opened would include read-write capabilities (provided that they have read-write permission to the map file). All other instances of the same map would be set to read-only.

The `Map:Save As` menu item copies only the currently *open* map and saves it under the new name you specify. Do this when you want to:

- Create multiple maps with similar characteristics.
- Create a specific map for each user. By copying one map, they all initially have similar or identical characteristics; although as changes are made, the maps will diverge.
- Create a writable version of the current map if the currently open map is set to read-only.

See the NNM online help if you need more information. See also the *ovw* reference page in NNM's online help (or the UNIX manpage) for information about command line options for copying maps.

The new map is not automatically opened. You can open the new map by selecting the `Map:Open` menu item and selecting the map name that you specified.

Copy the default map now and open the copy before you start making customizations. You can always return to your default map if you make a mistake.

## Your Map Strategy

Refer to your worksheets from Chapter 4, *Planning Your NNM Configuration*, where you identified which maps were needed by your organization. This chapter leads you through the map customization process. Think about how you might leverage your work from one map to another to keep this process as simple as possible. For example, first

make customizations that are common to all required maps. Then make multiple copies of that map before making the finishing touches for each department.

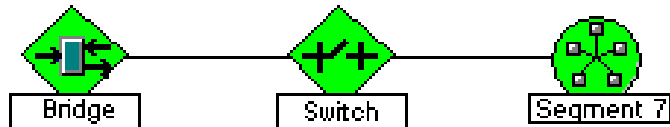
You can set a different Home submap for each user of the map, if you wish (page 273).

## Controlling the Display of Devices Attached to Switches or Bridges

NNM has two methods for displaying the devices that are directly attached to ports on switches or bridges:

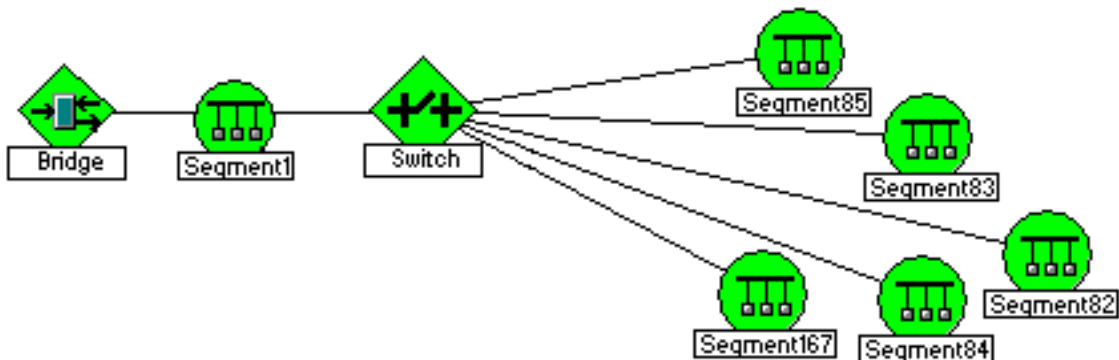
1. Each attached device is presented as part of a star configuration. You double-click on the star segment icon that is attached to a switch or bridge to display all attached devices on one submap. The star segment presentation is the recommended choice. Your map is less cluttered and easier to use when troubleshooting problems.

**Figure 8-1** SegRedux On



2. Each attached device is presented as a separate bus segment. You can check the status of each device on its own separate bus segment.

**Figure 8-2** SegRedux Off



## Turning On Star Configuration for Attached Devices

If you installed NNM for the first time, devices that are directly attached to ports on switches or bridges are automatically displayed in the star configuration. If you *upgraded* NNM from a previous version, you need to take the following steps in order to update your map and NNM's databases so that devices directly attached to ports on switches or bridges are displayed in the star configuration.

1. Log in as:
  - *Windows*: Administrator
  - *UNIX*: root
2. Open the `netmon` local registration file in the ASCII editing software of your choice:
  - *Windows*: `install_dir\lrf\netmon.lrf`
  - *UNIX*: `$OV_LRF/netmon.lrf`
3. Add the text shown in bold:

```
netmon:netmon:
OVs_YES_START:ovtopmd,pmd,ovwdb:-P -k segRedux=true:OVs_WELL_BEHAVED:15:PAUSE::
```

See the `netmon` reference page in NNM's online help (or the UNIX manpage); read about the `-k` keyword parameter and the `segRedux` keyword argument for more information.

4. Turn off the `netmon` service (background process). At the command prompt, type: **`ovstop -c netmon`**
5. Force NNM to acknowledge the change. At the command prompt, type the following two commands:
  - *Windows*:

```
ovdelobj install_dir\lrf\netmon.lrf
ovaddobj install_dir\lrf\netmon.lrf
```

- *UNIX*:
 

```
$OV_BIN/ovdelobj $OV_LRF/netmon.lrf
$OV_BIN/ovaddobj $OV_LRF/netmon.lrf
```

6. Turn on the `netmon` service (background process). At the command prompt, type: **`ovstart -c netmon`**

NNM now displays all devices that are directly attached to the ports on a switch or bridge in one star segment submap.

## Turning Off Star Configuration for Attached Devices

If you want to turn off this feature, you need to take the following steps in order to update your map and NNM's databases.

1. Log in as:
  - *Windows:* Administrator
  - *UNIX:* root
2. Open the netmon local registration file in the ASCII editing software of your choice:
  - *Windows:* `install_dir\lrf\netmon.lrf`
  - *UNIX:* `$OV_LRF/netmon.lrf`
3. Change the *true* to *false* (or delete the text shown in bold):

```
netmon:netmon:
OVs_YES_START:ovtopmd,pmd,ovwdb:-P -k segRedux=true:OVs_WELL_BEHAVED:15:PAUSE::
```

See the *netmon* reference page in NNM's online help (or the UNIX manpage); read about the `-k` keyword parameter and the `segRedux` keyword argument for more information.

4. Turn off the netmon service (background process). At the command prompt, type: **ovstop -c netmon**
5. Force NNM to acknowledge the change. At the command prompt, type the following two commands:
  - *Windows:*

```
ovdelobj install_dir\lrf\netmon.lrf
ovaddobj install_dir\lrf\netmon.lrf
```

- *UNIX:*

```
$OV_BIN/ovdelobj $OV_LRF/netmon.lrf
$OV_BIN/ovaddobj $OV_LRF/netmon.lrf
```

6. Clean up NNM's databases to update the configuration information. At the command prompt, type:

## Controlling the Display of Devices Attached to Switches or Bridges

- *Windows:* `ovtopofix -1`
- *UNIX:* `$OV_BIN/ovtopofix -1`

See the *ovtopofix* reference page in NNM's online help (or the UNIX manpage); read about the `-1` parameter for more information.

7. Turn on the `netmon` service (background process). At the command prompt, type: `ovstart -c netmon`

NNM displays a separate bus segment for each device directly attached to a port on a switch or bridge.

## Giving Your Network Symbols Meaningful Names

You can easily provide meaningful names for your network symbols, rather than their IP addresses; for example, Finance, R&D, or Manufacturing. Simply make additions to the list in the following file:

- *Windows:* \WinNT\system32\drivers\etc\networks
- *UNIX:* /etc/networks

Follow the directions at the top of this ASCII file. Make a list of all your network IP addresses and the corresponding names that would be most meaningful to your team. To force NNM to acknowledge the change, at the command prompt type:

```
ovstop netmon  
ovstart netmon
```

Once NNM synchronizes, the new names appear on your maps. Your changes to this file are preserved when applying NNM patches or upgrades.

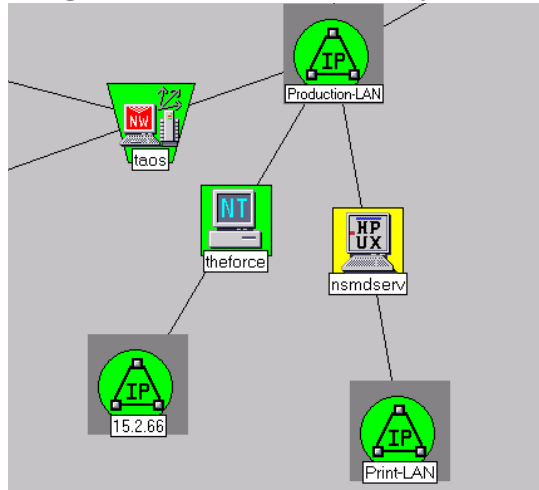
---

**NOTE**

This change is visible from remote consoles and web access after the map is reloaded or refreshed on each system.

---

**Figure 8-3**      **Meaningful Names for Network Symbols**

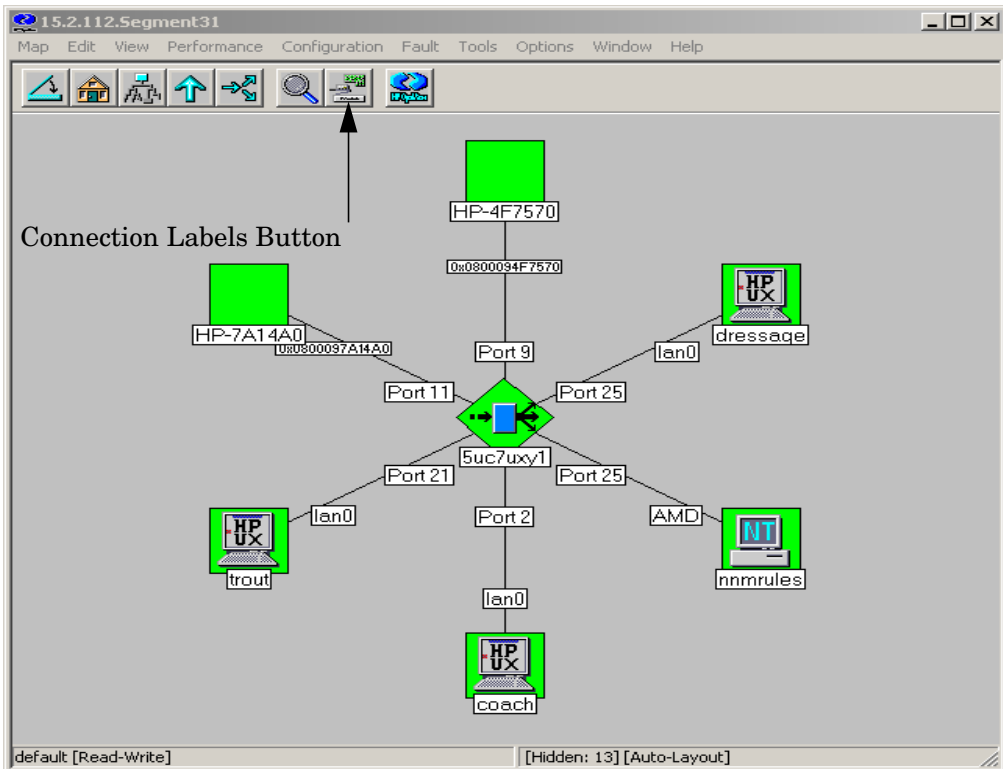




## Turning Connection Labels on or off

Connection labels indicate the port used to connect one network interface to another. Use either the `View:Show Connection Labels` menu item or the `Connection Labels` button to turn this feature on or off. On a Windows operating system you can control the connection label feature through the `View` tab contained in the `Map:Properties` or `Submap:Properties` pull-down menus.

**Figure 8-4** Show Connection Labels Enabled



NNM labels each end of a connection unless one end is a container object such as a segment, network, or location. Connection labels are created from information made available by equipment manufacturers. Since

## Turning Connection Labels on or off

devices return information in varying formats, connection labels vary from device to device. If NNM cannot read or understand the information from a device, it will not label the connection.

For devices that do not respond to SNMP requests, the connection label usually defaults to the IP address of the node. One common cause of devices not responding to SNMP requests is a mismatched community name. If you suspect that your GET-Community names are not matching up with the settings on your management station, see “GET- and SET-Community Name and SNMP Port Issues” on page 128.

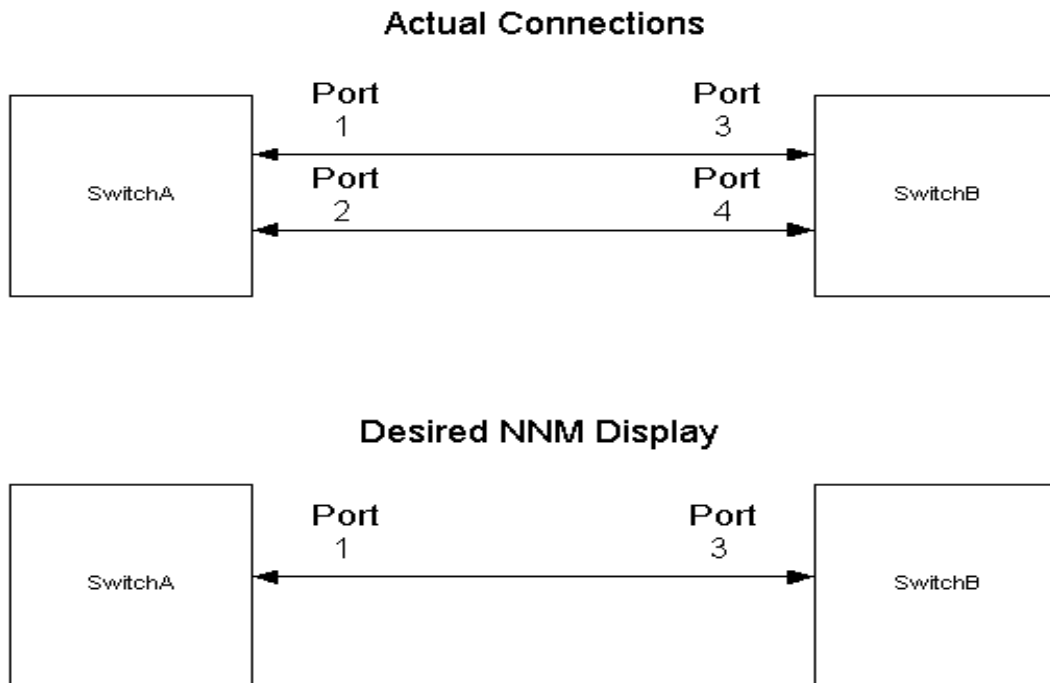
The connection label depends on information gathered during discovery and may be updated during a configuration poll. Changes on the devices may not appear on the connection label until the next configuration poll. To force an update of the connection labels, at the command prompt, type **ovtopofix -U**.

---

## Configuring Trunking and Meshing

Configuring two or more ports between two network devices into a single group is referred to as **port trunking**. This is done to increase the available bandwidth between two devices. For example, look at the actual connections diagrammed in Figure 8-5. The available bandwidth between switches A and B increases when ports 1 and 2 on switch A and ports 3 and 4 on switch B are connected in a trunk configuration.

**Figure 8-5** A Port Trunking Example



NNM is unable to detect port trunking and meshing automatically. For example, NNM will move devices around on your map depending on which port in the trunk or mesh configuration it last heard the device. To control NNM's representation of this topology, you can enter port trunking configuration information into the following file:

- *Windows:* `install_dir\conf\netmon.equivPorts`
- *UNIX:* `$OV_CONF/netmon.equivPorts`

To specify a trunk between two switches, add a line in the `netmon.equivPorts` file for each switch participating in the trunk. The `netmon.equivPorts` file consists of parameters in the format `host:port-spec:port-spec:[port-spec]` where `host` is an object identifier such as a host name, IP address, or MAC address, and `port-spec` is a combination of `group_number`, `port_number`.

When labeling any of the ports participating in a trunk relationship, NNM labels the object with the first `port-spec` parameter from the `netmon.equivPorts` device line entry.

For example, to achieve the results shown in Figure 8-5, the `netmon.equivPorts` file would have the following entries.

```
SwitchA:1:2  
SwitchB:3:4
```

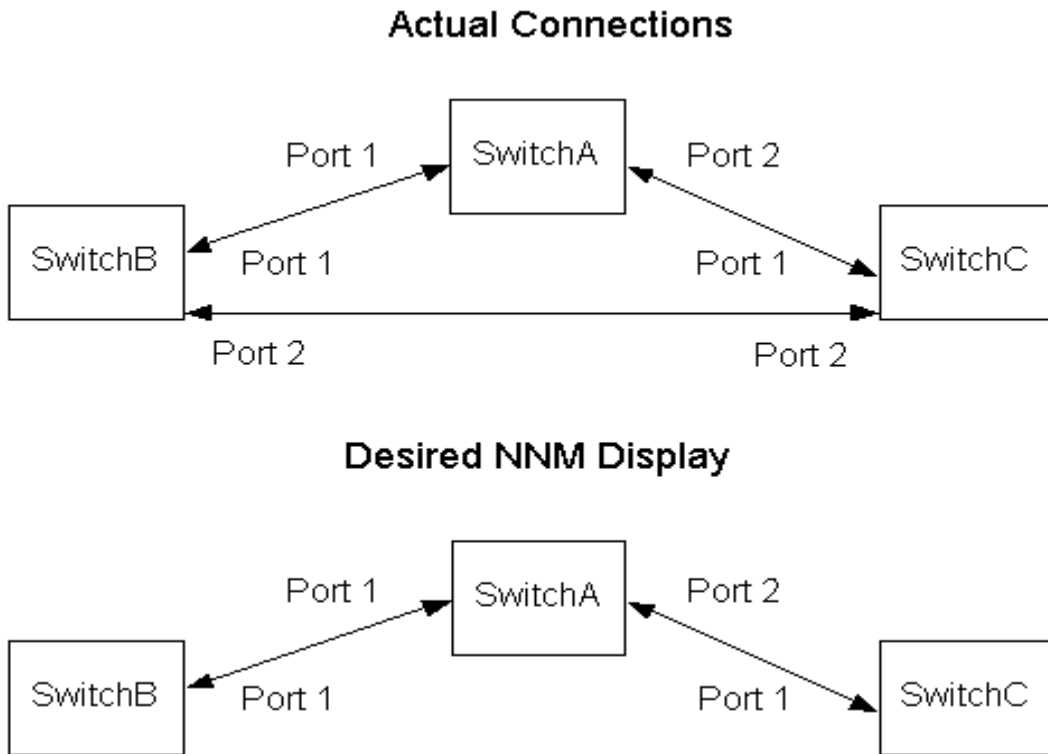
To use the new configuration, you must restart `netmon` using one of the following methods:

- From the command line, type `ovstop -v netmon` then type `ovstart -v netmon`.
- From the command line, type `xnmpolling -event`.

See the `netmon.equivPorts` reference page in NNM's online help (or the UNIX manpage) for more information

**Meshing** involves connecting devices on the network using multiple paths. Meshing is used when devices are critical and redundant network connectivity is required. For example, in Figure 8-6 the *Actual Connections* diagram shows Switch B port 2 and Switch C port 2 connected to each other, providing a redundant path between them other than just the path through Switch A.

**Figure 8-6** A Meshing Example



To control NNM's representation of this topology, enter meshing configuration information into the following file:

- *Windows:* `install_dir\conf\netmon.equivPorts`
- *UNIX:* `$OV_CONF/netmon.equivPorts`

To specify a mesh, add a line in the `netmon.equivPorts` file for each switch participating in the mesh. The `netmon.equivPorts` file consists of parameters in the format `host:port-spec:port-spec:[port-spec]` where `host` is an object identifier such as a host name, IP address, or MAC address, and `port-spec` is a combination of `group_number`, `port_number`.

When labeling any of the ports participating in a mesh relationship, NNM labels the object with the first `port-spec` parameter from the `netmon.equivPorts` device line entry.

For example, to achieve the results shown in Figure 8-6, the `netmon.equivPorts` file would have the following entries:

```
SwitchB:1:2
```

```
SwitchC:1:2
```

To use the new configuration, you must restart `netmon` using one of the following methods:

- From the command line, type **`ovstop -v netmon`** then type `ovstart -v netmon`.
- From the command line, type **`xnmpolling -event`**.

See the `netmon.equivPorts` reference page in NNM's online help (or the UNIX manpage) for more information

## Establishing Submap Persistence Settings

The persistence settings allow you to control the number of submaps that are retained in memory (RAM) so that they are available at all times, rather than generated upon demand. There are several of things to keep in mind when making your choices:

- Persistent submaps display faster.
- The greater the number of persistent submaps, the greater the need for RAM on the management station.
- Only those submaps that are persistent are included when a map snapshot is created.
- Only those submaps that are persistent are included in the initial scoping pane list for NNM's web access.
- You can set persistence by logical level of network structure (such as internet level, network level, or segment level).
- You can establish a persistence filter that specifies making persistent all submaps that contain certain devices, that you specify (page 216).
- Any time that you make an editing modification directly to a submap, that submap and all related parent submaps up to the internet level become persistent; for example, turn off auto-layout, set window geometry, add a background graphic, or add an executable symbol.

---

### NOTE

If you add background graphics to your submaps, they automatically become persistent. Keep in mind that graphic files are quite large and may significantly increase the amount of memory required to keep the submap in RAM.

---

NNM's default setting for this feature:

- *Windows*: Only the Internet level and the containment realm of the map are persistent.
- *UNIX*: All levels of the map are persistent.

---

**NOTE**

---

The selection that you make affects remote consoles and the web display after the map is reloaded or refreshed on each system.

## By Logical-level of Your Network's Structure

If you choose to set the persistence by logical level, select the `Map:Properties` menu item, select the `Applications` tab, and double-click the `IP Map` entry. The level to which submaps should be persistent can be one of the following values:

`Unset` *Windows:* Same as Internet Level.

*UNIX:* Same as All Levels.

`All Levels` Specifies all submaps as persistent.

`Segment Level and Higher` Specifies that Segment-, Network-, and Internet-level submaps are persistent.

`Network Level and Higher` Specifies that Network- and Internet-level submaps are persistent.

`Internet Level` Specifies that Internet-level submaps are persistent.

## By Presence of Specific Devices

A persistence filter describes which specific network devices should have their submaps retained in memory (RAM) at all times.

This may be useful if you want to use the snapshot feature (see NNM's online help) as part of your normal business procedures and need to make sure the snapshots always include information about specific critical devices. It may also be a useful way to keep the number of submaps retained in memory (RAM) to a minimum when you are using an HP OpenView program that requires the submaps containing certain devices to remain present in RAM.

The persistence filter is defined in the `filters` file (the same file that contains any discovery filters, DHCP filters, etc.):

- *Windows:* `install_dir\conf\C\filters`
- *UNIX:* `$OV_CONF/C/filters`



NNM includes a predefined persistence filter section for your convenience as a starting point, or you may define your own. For detailed information about how to develop a filter, refer to *A Guide to Scalability and Distribution*.

After you create your persistence filter:

- Select the `Map:Properties` menu item, select the `Applications` tab, and double-click the `IP Map` entry.
- Click on `Persistence Filter` and type the name of the section in NNM's `filters` file that you just created.

## Controlling Which Devices Appear on the Map

You can control which devices are visible on a particular map. It is important to remember that all maps derive their information from the same database, so *deleting* an object (specific device) from the database eliminates it from all maps (until the next discovery polling interval). Rather than using delete, consider the following choices:

- **Map Filter:**

A map filter allows you to remove nodes from view on the map. You write the map filter using Boolean logic and place it in the `filters` file with all the other NNM filters. You can create multiple map filters and assign a unique filter to each map.

- **Hide Feature:**

The hide feature enables you to modify which group of symbols is displayed on any submap in an interactive way (without needing to write a filter using Boolean logic).

---

**NOTE**

These changes are visible from remote consoles and web access after the map is reloaded or refreshed on each system.

---

### Creating a Map Filter

A map filter allows you to remove certain nodes from view on the map while continuing to monitor them and receive alarms from them in the event/alarm portion of NNM. Map filters are defined in the `filters` file (the same file that contains any discovery filters, DHCP filters, and so on):

- *Windows:* `install_dir\conf\C\filters`
- *UNIX:* `$OV_CONF/C/filters`

See *A Guide to Scalability and Distribution* for information about creating filters.

You can create multiple map filters and assign a unique filter to each map. See the `filters` file for predefined examples as a starting point. Your filter can be applied/changed at any time by opening the map, selecting `Map:Properties`, selecting the `Applications` tab, and opening the configuration dialog box for `IP Map`.

### Testing Your Filter

See *A Guide to Scalability and Distribution* for information about testing and troubleshooting your filters.

### Using the Hide Feature

The hide feature allows you to customize individual submaps as well as all submaps in a map. For example, you may wish to hide everything except printers to create a special set of submaps specifically for monitoring printers, or hide everything except routers to create a special submap specifically for monitoring routers. Because all maps derive their information from the same database, if you delete all instances of any symbol, the underlying object in the database is also deleted. The hide feature enables you to modify the group of symbols displayed on any submap in an interactive way (without needing to learn Boolean logic in order to write a filter).

---

#### NOTE

If the submap is transient, it becomes persistent when you hide a symbol. (Use the map filter if you wish to hide symbols without effecting persistence.)

---

If devices are hidden on a particular submap:

- `[Hidden:#]` appears on the right side of the status bar giving notice about how many devices are hidden (`#` = number of hidden objects).
- There is no way to determine which devices are hidden, only how many are hidden.
- If you wish to show a hidden object on a particular submap, you must show all hidden objects. There is no way to specify which hidden objects to show.

## Controlling Which Devices Appear on the Map

To hide a symbol, select the symbol or symbols that you wish to hide, select `View:Hidden Objects`, and select `Hide Selected From This Submap` or `Hide Selected From All Submaps`. You can use the `Find` feature in partnership with the `Hide` feature to hide groups of symbols such as all routers.

### Testing and Troubleshooting

If you choose to hide a symbol from all submaps but change your mind later, to undo the action you must show ALL hidden symbols on ALL submaps.

To display hidden symbols on individual submaps, open each affected submap and select `View:Hidden Objects->Show Hidden On This Submap`.

To display hidden symbols on all submaps, open any submap and select `View:Hidden Objects->Show Hidden On All Submaps`.

### Backing Up Your Efforts

The next time NNM's backup process runs, your modifications will be included.

---

## Changing/Adding Object Attribute Fields

You may wish to modify or add object attribute field definitions so that your team can search on custom criteria when resolving network problems or so that a new application can manage the object.

---

### NOTE

You can only add fields, you cannot delete them. Adding fields does not affect submap persistence.

---

An NNM object represents a logical or physical entity or resource, or a group of such logical or physical entities or resources that exist in a network environment. In NNM, an object usually represents any particular thing of interest for the purpose of doing network or systems management. An object can represent a physical item (such as a PC, a gateway, a router, or an interface card) or it can represent a logical item (such as a group of PCs, all 486 PCs, or all nodes in a single department). An object is represented on the map graphically by a symbol.

An object is constructed of individual characteristics that are called attributes. Attributes include the name of the object and various characteristics that are used to classify the object. An object, for example, might have the name IP Hostname and an attribute such as isDevice.

To view the attributes assigned to any object, select any symbol that represents the object and then display the `Edit:Object Properties` dialog box. A list of attribute categories is displayed, such as Capabilities, General Attributes, and application-specific categories such as IP Map (an application included in NNM). Select any category to display its list of attribute names and values.

The attribute names represent fields in the NNM object database, and the data in the fields are the object's attribute values. Possible modifications/additions to the information in the object database:

- Change the current value in an attribute field through the object properties' `General Attributes` dialog box.
- Change a symbol's *type* in order to switch to a different set of attribute fields.

- Add underlying definitions of attribute fields by editing field registration files (FRF).

These changes are visible from remote consoles and NNM's web interface after the map is reloaded or refreshed on each system.

---

**TIP**

The following *analytical data* directories are *not* included in NNM's backup scripts:

- *Windows:*

```
install_dir\backgrounds
install_dir\bitmaps
install_dir\www\htdocs\bitmaps
install_dir\www\registration
```

- *UNIX:*

```
$OV_BACKGROUNDDS
$OV_BITMAPS
$OV_WWW/htdocs/bitmaps
$OV_WWW_REG
```

If you make changes or additions to map backgrounds, bitmap files, or web registration files ensure that your new files are properly backed up. See “Backup/Restore to Protect Your Investment of Time” on page 149 for more information.

---

## Changing the Value in an Attribute Field

You can change the description of an object by modifying the values assigned to its attributes (characteristics) in the General Attributes dialog box. See NNM's online help for more information.

To change object attribute values:

- *Windows:* Select a symbol; then select Edit: Object Properties. Alternatively, right-click the symbol and select Object Properties from the pop-up menu. The values in the General Attributes dialog box can be modified. To edit these values, click on the attribute name.

- *UNIX*: Select a symbol; then select `Edit:Object Properties`. Alternatively, click the mouse button on a symbol and select `Object Properties` from the pop-up menu. The values in the `General Attributes` dialog box can be modified. To edit these values, click on the attribute name.

For example, clicking on `isSNMPSupported` will toggle the value between `TRUE` and `FALSE`; clicking on `SNMPAgent` will pop-up a dialog box from which you can choose the SNMP agent.

---

**NOTE**

This change is visible from remote consoles and the web interface after the map is reloaded or refreshed on each system.

---

## Changing Symbol Type to Switch Attribute Sets

You can assign a new set of attributes to an object by changing the symbol type of the object.

---

**NOTE**

This step adds fields, but does not delete or change values in existing fields.

---

*Windows*: Right-click a symbol and select `Change Symbol Type` from the pop-up menu.

*UNIX*: Click mouse button 3 on a symbol and select `Change Symbol Type` from the pop-up menu.

---

**NOTE**

This change is visible from remote consoles and the web interface after the map is reloaded or refreshed on each system. Changing the symbol type forces the submap to become persistent.

---

See NNM's online help for more information about changing symbol type. See also the *OVwRegIntro* reference page in NNM's online help (or the UNIX manpage) for more information.

---

**NOTE**

The `ipmap` service automatically assigns a symbol type and label based upon information it receives from the `netmon` service. This means that if you modify the symbol label or symbol type of an object that is managed by NNM, your changes will be replaced by the values assigned by NNM the next time configuration polling runs.

You can turn off this `ipmap` capability by setting the environment variable `IPMAP_NO_SYMBOL_CHANGES=TRUE`. See the `ipmap` reference page in NNM's online help (or the UNIX manpage) for information about the `-u` parameter.

Alternatively, you can modify the appropriate file in the `oid_to_sym_reg` directory to specify that a different symbol be associated with a specific *type* of device (determined by `sysObjectID`). See the `oid_to_sym` reference page in NNM's online help (or the UNIX manpage) for more information. See "Changing All the Symbols for a Particular Device" on page 623 for a complete example.

---

## Adding Attribute Fields in the Object Database

If you want to add a specific attribute field that is not yet defined in the object database, you simply create or modify the field registration files (FRF):

- *Windows:* `install_dir\fields\C\*.*`
- *UNIX:* `$OV_FIELDS/$LANG/*`

See NNM's online manual, *Creating and Using Registration Files*, for complete information about changing fields or adding fields.

See the *OVwRegIntro* reference page in NNM's online help (or the UNIX manpage) for more information.

---

**TIP**

Set the `Locate` and `General` flag for the field when you define it. `Locate` enables your new field to appear in the `Edit:Find->Object by Attribute` dialog box. `General` enables the field to show up in the `Object Properties, General Attributes` category.

---



After you have modified the field registration files:

1. Display a list of currently running sessions. At the command prompt type:

```
ovstatus -v ovuispmd
```

You may want to print out this information so that you have the hostnames and session IDs for restarting the sessions later. You may wish to notify those users who are currently accessing NNM to warn them about the shutdown.

2. Stop all sessions. At the command prompt, type:

```
ovstop -c ovuispmd
```

(This terminates any sessions and stops only the `ovuispmd` service--the maps are all closed, but your network is still being monitored by NNM.)

3. Force NNM to update the field definitions in the object database. At the command prompt, type:

```
ovw -fields
```

4. To start new NNM sessions, at the command prompt, type:

```
ovstart -c ovuispmd
```

5. Open NNM's user interface to begin one or more sessions.

The field registration files are parsed for correctness and checked for errors which are reported to:

- *Windows*: The error message pop-up Console window. Messages are always appended to the end of the Console file. It is a good idea to clear the contents of this file (`Edit:Clear`) each time that you see it.
- *UNIX*: `stderr`

Close the NNM interface and reopen the maps to view your changes (see "Closing All Current Sessions" on page 305).

---

**NOTE**

These changes are visible from remote consoles and web access after the map is reloaded on each system.

---

## Making the Maps Look Like Your World

Review the worksheet from Chapter 4, Planning Your NNM Configuration, where you identified the maps that your organization needs and the network devices that need to be present on each map. Now customize the look of each map to meet the needs of your team. When you edit a map, your changes affect only the *open* map. If you want the same customizations on multiple maps, make the changes once, then copy the modified map. A map can be customized in the following ways:

- By controlling the placement of the symbols on your map. By default, NNM controls how symbols are placed on your submaps, but you can take control of how symbols are placed on individual submaps. You can also turn off auto-layout for the whole map.
- By adding your own submaps:
  - *Partitioned Internet* submaps (child submaps at the Internet-level) to break the Internet symbols into smaller groups of information, such as by country, city, building, department, or floor by using container objects (location symbols). These changes (*and these changes only*) can be copied to other maps using the Map:Export and Map:Import feature.
  - *Child* submaps (at the Network-, Segment-, or Node-level) to add more information to an explodable symbol.
  - *Meta-connection* submaps automatically created by NNM whenever there are multiple connections between two symbols. Double-click on the connecting line to explode a submap showing each connecting-line symbol.
  - *Independent* submaps to create your own custom hierarchy of submaps that is independent of the hierarchy automatically created.
- By adding background graphics such as map images, site-specific images, or floor plans (“Background Graphics” on page 246). Use \*.GIF files to ensure that the background graphics are visible over web access.

---

**NOTE**

These changes are visible from remote consoles after the map is reloaded or refreshed on each system. These changes are immediately visible in NNM on the web.

---

## Controlling the Placement of the Symbols on Your Map

NNM's auto-layout controls the placement of symbols on the maps (see "Automatic Discovery and Layout" on page 38 for more information about how auto-layout works). If you want to make changes to the placement of symbols, you must first turn off auto-layout. Auto-layout can be turned On/Off for a single submap or for all submaps in the open map. When auto-layout is turned off, the [Auto-Layout] message at the bottom of each affected submap disappears, and the New Object Holding Area appears across the bottom of the submap. Any time a new object is discovered, a map symbol is moved to another submap, a bridge or hub is moved within a network submap, or a hidden object is set to show on your network, the affected symbols will show up in the New Object Holding Area to await your decision about placement.

---

**TIP**

When accessing NNM over the web, there is no New Object Holding Area. However, the status bar notifies you if objects are currently in the New Object Holding Area.

---

### Auto-Layout Algorithms (Default Layout Rules)

The layout algorithm for a submap is set when the submap is created. If an application (such as NNM) creates the submap, it specifies a layout algorithm. NNM's default layout algorithm for a submap is based upon the symbol type of the parent object and cannot be changed after the submap is created.

The possible layout algorithms are:

- None. Symbols are arranged by the user or are left in the New Object Holding Area.

- Point To Point. Symbols are arranged as an arbitrarily interconnected set of nodes and connections.
- Row/Column. Symbols are arranged in rows and columns.
- Bus. Symbols are arranged along a snake-like backbone representing the linear array of nodes on a segment.
- Star. Symbols are arranged in a star consisting of a circle and a center symbol. You can set the star center using the symbol pop-up menu.
- Ring. Symbols are arranged in a circle.

If automatic layout is enabled:

- [Auto-Layout] appears on the right side of the status bar.
- When a new symbol is added to the submap, NNM places the symbol on the submap according to the assigned layout algorithm.
- Although you can move the symbols in the submap, all symbols in the submap snap back to their original position when the `ipmap` service checks for changes.

By default, auto-layout is turned on. If you turned it off and change your mind, to turn auto-layout on:

- *Windows:*

For all submaps, select `Map:Properties` and access the `View` tab.

For one submap, select `Map:Submap->Properties` from that submap and access the `View` tab. Select the `Auto Layout` field.

- *UNIX:* Select either:

`View:Automatic Layout->On For All Submaps`

`View:Automatic Layout->On For This Submap`

### Do-It-Yourself Layout

If you want to create a customized placement (for example, to place symbols on a background graphic that shows their geographical location), you need to turn off automatic layout for the submap you want to customize.

---

**NOTE**

---

If the submap is transient, it becomes persistent when you turn off auto-layout.

If automatic layout is disabled:

- A `New Object Holding Area` is created in the lower portion of the submap. Symbols in the `New Object Holding Area` are shown without their connections. Connections are automatically added when you remove the symbol from the holding area and place it on the map.
- Symbols added to the submap are placed in the `New Object Holding Area`. You can move symbols from the `New Object Holding Area` to the submap.
- You can move symbols within the submap window by dragging and dropping them with the mouse.
- You cannot move a symbol from the submap to the `New Object Holding Area`.
- You can arrange symbols in the submap according the assigned layout algorithm by selecting `View:Redo Layout`.
- When hidden symbols are unhidden, they are placed in the `New Object Holding Area`.

To disable auto-layout:

- *Windows:*

For all submaps, select `Map:Properties` and access the `View` tab.

For one submap, select `Map:Submap->Properties` from that submap and access the `View` tab. Deselect the `Auto Layout` field.

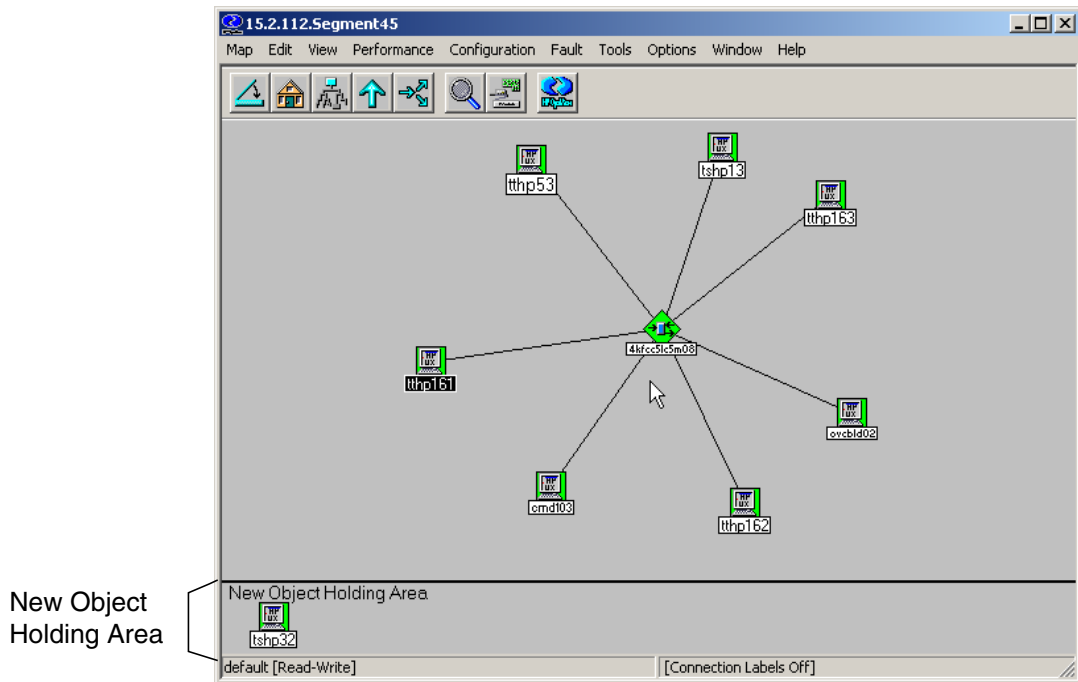
- *UNIX:* Select either:

`View:Automatic Layout->Off For All Submaps`  
`View:Automatic Layout->Off For This Submap`

### New Object Holding Area

A New Object Holding Area (Figure 8-7) appears on the lower portion of each submap window if the submap has no layout algorithm or if the assigned layout algorithm is disabled (automatic layout is turned off). Symbols in the New Object Holding Area are shown without their connections.

**Figure 8-7** The New Object Holding Area



### Adding Your Own Submaps

You can create submaps that show a particular view of some aspect of your network. For example, you may create one submap to show all the nodes on a particular network segment, and create another submap that shows all the software subsystems of a particular node.

## Child Submap or Independent Submap?

The method you use to create a new submap determines whether the submap is a *child submap* or an *independent submap*.

- A *child submap* is related to a parent object. To create a child submap, double-click on an explodable symbol. If no child submap exists, you are prompted to create one (for example, when you create a new container object).
- An *independent submap* is created from the `Map:Submap->New` menu item. The submap has no associated parent object and is independent of any existing hierarchy of submaps. To display an independent submap, select a submap entry in the `Edit:Find->Submap` dialog box.

---

### TIP

You cannot open the independent submap by double-clicking on a symbol unless you manually add a symbol to the root-level submap that invokes your independent submap.

---

An independent submap can be used as the beginning of a new submap hierarchy. You must create any child submaps starting from the independent submap to build an additional submap hierarchy into the map. NNM cannot automatically generate these child submaps for you. NNM also cannot update your independent hierarchy as new nodes are discovered or changes occur to the connections.

The map must be open with read-write access before you can create a submap.

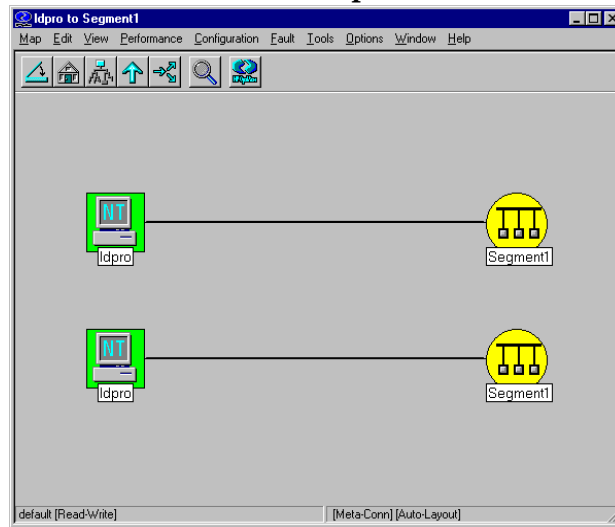
## Meta-Connection Submaps (for multiple connections)

If two devices represented by symbols on your map communicate over *multiple* connections, you can double-click upon the connecting line to display a meta-connection submap. The meta-connection line (symbol) on your submap looks the same as a regular line on your map (connection symbol).

NNM does not display multiple lines between the two objects; instead, each individual connection between the two symbols is displayed on the line's meta-connection submap. NNM automatically creates a meta-connection submap when you (or an application) add a connection

symbol between two symbols in a regular submap that already have a connection. You can add an unlimited number of connections between two symbols in a regular submap. Each of these connections is automatically added to the meta-connection submap with the two symbols as end points.

**Figure 8-8** Meta-Connection Submap: Multi-Interface Connections



A *meta-connection submap* displays the status of each connection between two symbols or a symbol and a backbone. Compound status colors from the meta-connection's submap are propagated to their parent submap differently as compared to regular submaps (page 260). The meta-connection symbol in the parent submap displays the compound status of the multiple connections in the meta-connection submap. Any unconnected objects in the meta-connection submap also contribute to compound status. However, the connected icon symbols in a meta-connection submap do not propagate their status. Their status is maintained by the symbols in the parent submap above the meta-connection submap. (Meta-connection submaps are not accessible over NNM's web interface.)

You can:

- Create a child submap from a meta-connection submap by double-clicking on any of the objects in the meta-connection submap. Therefore, the meta-connection submap can be a parent of other regular submaps.



- Select objects in the meta-connection submap.
- Add unconnected objects to a meta-connection submap.
- Add connections at the parent level submap. (You cannot add connections directly on a meta-connection submap.)
- Delete the meta-connection submap by deleting the last object from the meta-connection submap.

## Customizing the Internet Level of Your Network Map

As a result of the initial discovery process, NNM displays one Internet symbol on the root submap *and creates one Internet submap*. Within the Internet submap are symbols representing IP and IPX networks, and IP- and IPX-addressable gateways (routers). This section provides definitions and procedures that you can use when editing the internet level of your network map (for information about editing the network, segment, or node levels, see page 242). In order to understand your options, you need to know:

- What are container objects.
- What are partitioned internet submaps.
- What is the containment realm.
- How to expand the single Internet submap into multiple partitioned internet submaps.
- How to edit and maintain the partitioned internet submaps.

---

### NOTE

To manage IPX networks, you must be running NNM on a management station using a Windows operating system. If you are running NNM Advanced Edition on a management station using a UNIX operating system, you can configure a system that uses the Windows operating system to serve as a collection station that forwards IPX information to the UNIX system. This requires another copy of NNM Advanced Edition to be running on the system with the Windows operating system. See *A Guide to Scalability and Distribution* for information.

---

## Container Objects to Hold Part of Your Map

You can group the symbols on the Internet-level submap into logical submaps that reflect the physical or logical reality of your network and working style.

First, you add symbols called *container objects* to the Internet submap. The container objects allow you to create maps by location, building, department, or whatever makes sense to you. Each container object explodes into a new submap called a *partitioned internet submap*. Container objects can be either the Internet symbol (in the Network class) or any of the symbols in the Location class (see NNM's online help Display Legend for a quick look at your choices).

Partitioned internet submaps can contain gateway router symbols and subnet symbols. To maintain the map hierarchy, partitioned internet submaps are only allowed between the Internet-level submap and the Network-level submap. NNM maintains the connection between devices as you create the partitioned internet submaps.

## Partitioned Internet Submaps

The symbol of a container object opens a special submap called a partitioned internet submap. You can add symbols from the Location class and the Internet symbol from the Network class to any partitioned internet submap. Thus, you can create a hierarchy of partitioned internet submaps as many levels deep as you wish. For example, you may want to subdivide your management domain by continent, then by country, then by city, showing your IP and IPX networks, and IP and IPX-addressable gateways (routers) only at the city level.

The Internet submap always remains at the top of your hierarchy of partitioned internet submaps and is opened whenever the Internet symbol is selected from the root map.

You can create multiple levels of partitioned internet submaps to rearrange how your network map is displayed. NNM maintains the connections and monitors the devices placed within a container object as long as the container objects are on submaps between the Internet-level submap and any Network-level submaps. If you use container objects at other levels of the map, NNM cannot manage the devices that are contained in the submaps, so connections are lost and status is not updated.

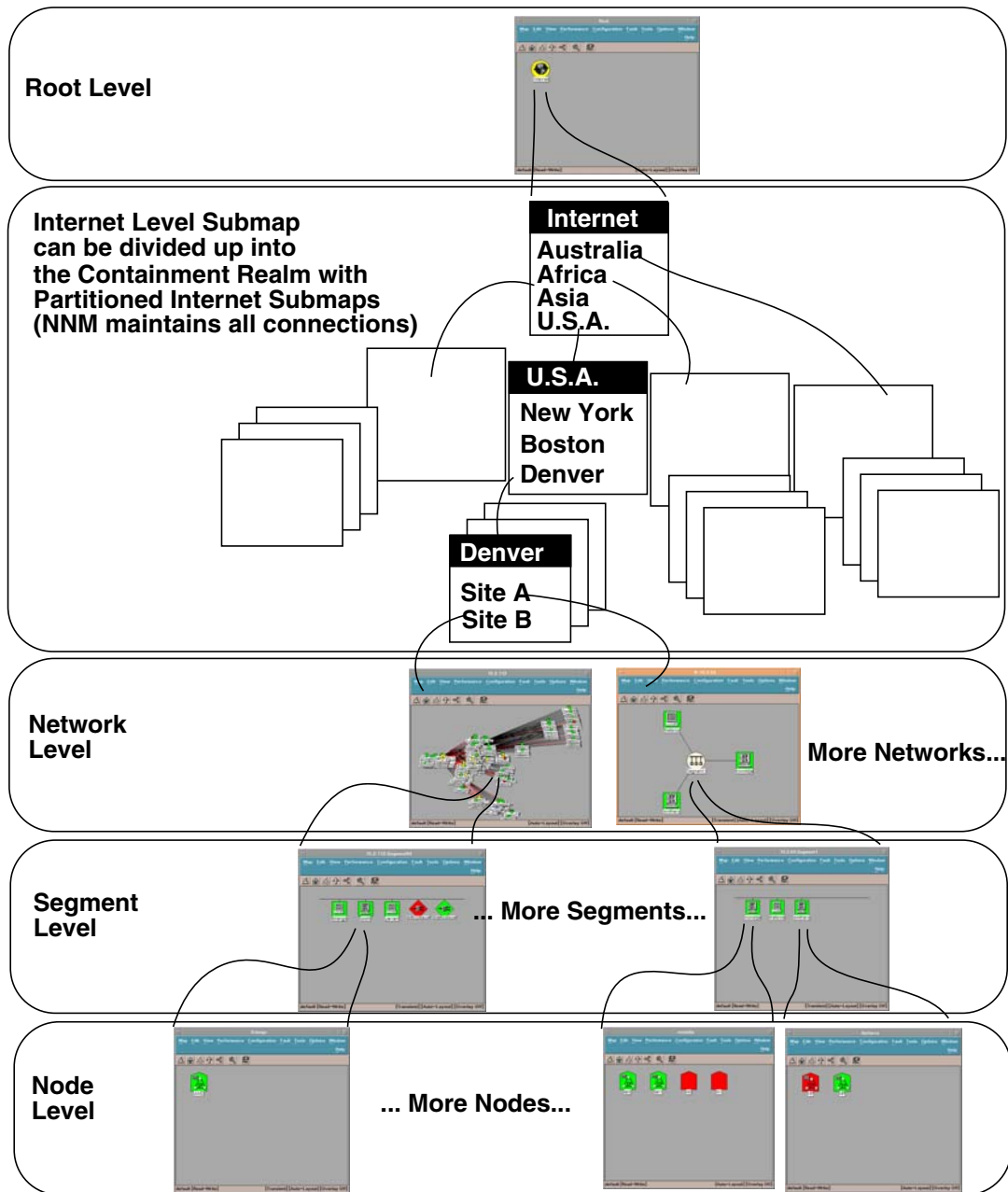
---

**NOTE**

Do not place a container object on the Root-, Network-, Segment-, or Node-level submaps. Only use container objects on submaps between the Internet- and Network-level submaps to ensure that NNM is monitoring your network devices.

---

Figure 8-9 Containment Realm: Partitioned Internet Submaps



## Creating a Partitioned Internet Submap

To create a partitioned internet submap, add the Internet symbol from the Network class or any of the symbols from the Location class to the Internet submap or any partitioned internet submap. The new symbol represents a special object, called a container object. Symbols of container objects open into a partitioned internet submap. You can then cut and paste IP and IPX network and gateway (router) objects from your Internet submap into your partitioned internet submap. NNM maintains all connections between your IP and IPX network and gateway (router) objects.

---

### NOTE

---

It is important to follow these directions *exactly*.

To add a container object to an Internet or partitioned internet submap:

1. Select the `Edit: Add Object` menu item. The `Add Object Palette` appears.
2. From the `Add Object Palette`, select either:
  - the `Network Class's Internet` symbol
  - the `Location Class` (any symbol)

Drag the symbol onto the Internet submap or the partitioned internet submap of your choice. The `Add Object` dialog box appears.

3. Specify your choices in the `Add Object` dialog box.
4. Click on `[OK]`. The newly added symbol resides on the submap, and an empty child submap (partitioned internet submap) for that symbol is automatically created. To open the new partitioned internet submap, double-click on the new symbol.

---

### NOTE

---

Until you add objects to your new partitioned internet submap, the container symbol displays an unknown status. In the new container symbol's `Object Properties` dialog box, `IP Map` is not listed as a configurable application because the `IP Map` service does not require any information about the container object.

5. Use `Edit:Cut` and `Edit:Paste` to place the desired IP and IPX network and gateway (router) objects into the new partitioned internet submap.

All submaps, objects, symbols, and connections that are contained under a cut-and-pasted symbol remain intact. Symbol behavior and status are maintained. For example, if the symbol you paste onto a map explodes into a child submap, that capability also exists for the new symbol.

### The Containment Realm

The Internet submap and the hierarchy of partitioned internet submaps make up the **containment realm**. You can populate a manually created partitioned internet submap by cutting and pasting IP and IPX network and gateway (router) symbols from the Internet submap to a partitioned internet submap. Only one symbol is allowed for each IP and IPX network and gateway (router) symbol; however that symbol can be placed in any location within the containment realm.

Within the containment realm, when you cut and paste these symbols the hierarchical and topological relationship between symbols is maintained. When you cut these symbols from the Internet submap and paste them on a partitioned internet submap, the hierarchical relationship is maintained down the map hierarchy. In other words, that symbol's child submaps along with all associated symbols and objects are also moved.

In addition to the hierarchical relationship, all topological relationships between symbols are maintained. NNM maintains the connections between symbols after they are pasted onto a partitioned internet submap by redrawing their connections to the symbol of the container object.

### Editing Guidelines

---

#### NOTE

You must follow these directions *exactly* in order to maintain the connections between devices on your map as you work and in order to maintain NNM's monitoring of your network devices.

---

This section discusses the types of editing operations you can perform in the containment realm *while maintaining the connections between symbols*.

To maintain connections between symbols in the containment realm:

Use `Edit:Cut/Edit:Paste` or drag-and-drop.

However, you must also follow these guidelines:

- You can cut and paste symbols between the Internet submap and any partitioned internet submap.
- You *cannot* cut from and/or paste symbols onto the Root submap or the Network, Segment, and Node submaps. In other words, cut/paste is only allowed on the Internet or containment realm level.
- You *cannot* cut and paste or drag and drop container symbols themselves. If you paste a container object on one of these submaps, the containment information is broken; that is, the symbol will no longer maintain its relationship to other symbols. Connections between that symbol and other symbols are also broken.

Once a cut and paste or drag and drop operation has been made between submaps, IP Map will need to resynchronize. You cannot perform any other operations until this resynchronization is complete.

### **Example of an Internet Submap**

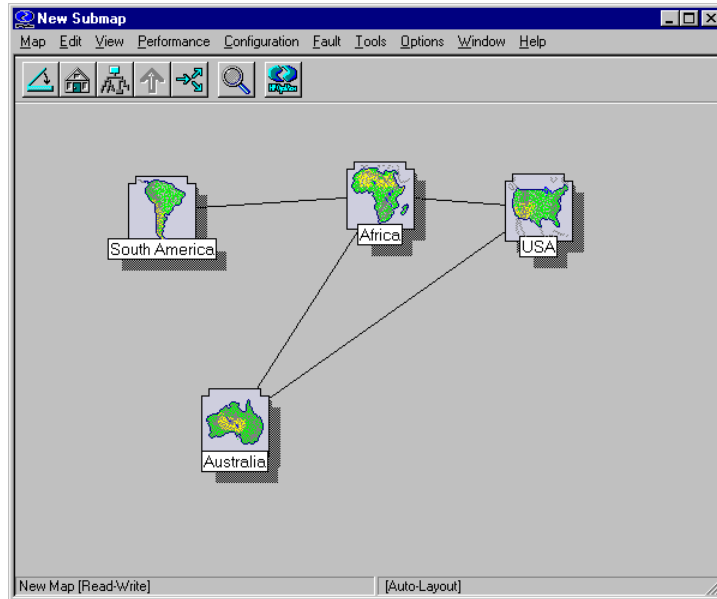
On the Internet submap, you can place a map of the world and add symbols that represent container objects of countries or continents. Each container object opens into its own submap. You can then cut symbols that represent your resources in a particular country or continent from the Internet submap and paste them into the appropriate partitioned internet submap.

The following figure shows an Internet submap in which four container objects were added. The symbols of the container objects represents Australia, U.S.A., South America, and Africa. In addition, all symbols of objects discovered by NNM have been cut from the Internet submap and pasted in lower-level partitioned internet submaps. NNM maintains the

connections between the symbols of discovered objects in the lower-level submaps. Also, NNM creates the proper connections between the symbols of the container objects.

**Figure 8-10**

**IP Internet Submap**



**Example of Partitioned Internet Submap**

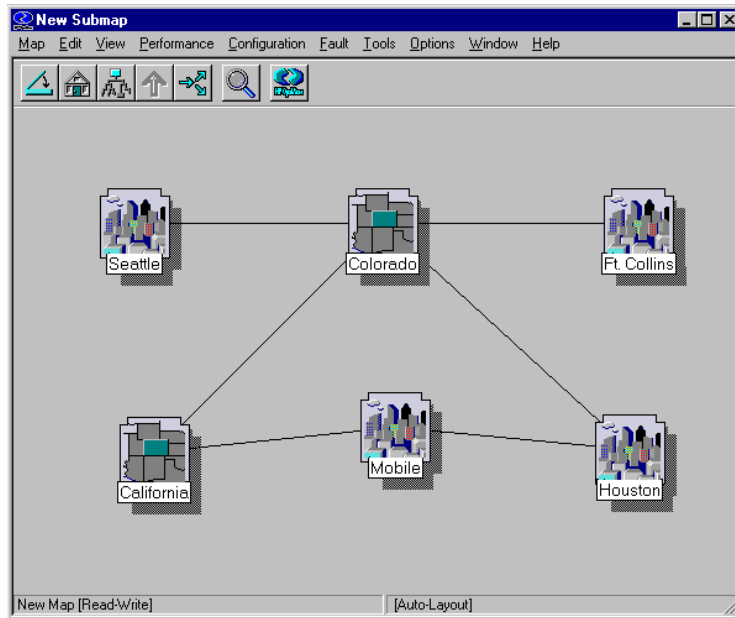
The following figure shows the U.S.A. submap. Six container objects have been created in this submap. This submap may also contain symbols representing IP and IPX networks or gateways. From this submap, you can double-click on any of the six container objects and open a



partitioned internet submap, which may display an additional partitioned internet submap containing other container objects or symbols that represent IP and IPX networks and gateways.

Figure 8-11

### Partitioned Internet Submap of the USA



### Maintaining the Containment Realm

As you customize your submaps, NNM tracks any editing changes you make to the open map. The Internet submap and all partitioned internet submaps make up the containment realm. NNM continues to manage objects that have been moved within the containment realm with the network monitoring process (`netmon`).

If new IP- and IPX-addressable resources are added to the network, NNM discovers these resources, creates objects for them, and adds a symbol for each object to the appropriate submap or submaps. NNM always places IP- and IPX-addressable gateways (routers) and networks that have been discovered on the original, highest-level Internet submap.

You can maintain the customized organization of your network map by distributing these symbols (“Editing Guidelines” on page 238) to the partitioned internet submap of your choice.

When creating multiple maps on one management station, using the `Map:Import` and `Map:Export` feature is an easy way to copy your *containment-realm customization* changes from map to map.

---

**CAUTION**

Always make an export file of any map before importing changes from another map. This enables you to undo the changes if you don't like what happened.

---

All your container objects as well as the objects that NNM automatically placed at the internet level are included. Changes to the lower-levels of the map (network/segment/node) and any executable symbols (page 265) are not included in the export/import nor are they affected in any way by the import/export. If you need multiple maps that include your customizations to *all* levels of the map, use `Map:Save As` instead.

---

**TIP**

The `Map:Export/Map:Import` feature *may* be useful if you are configuring NNM on a test machine and plan to move changes to a production machine at a later time. Make sure that exactly the same HP OpenView and 3rd party products are installed on both machines to ensure a successful map export/import operation. See NNM's online help for more information about Map Import/Map Export. If you need to move customization at all levels of your map, rather than just at the Internet and containment realm levels, use NNM's backup and restore feature instead (page 149).

---

## Customizing Network-, Segment-, or Node-Level Submaps

Editing the lower-level network submaps enables you to distribute resources over your map to better match how your network is organized. You can:

- Copy or move symbols within or between submaps.
- Add symbols to your submaps.

## Copying or Moving Symbols

Copying or cutting and pasting are convenient ways to edit your network map. The cut and paste operation is the means for changing a symbol's location from one submap to another, and is essentially a move operation. You can cut and paste multiple symbols simultaneously.

If you cut and paste symbols at the network, segment, or node levels, NNM will *not* maintain the previous connections between these symbols. Therefore, after cut and paste operations, you must add a connection between the desired symbols. Use the dialog box accessed through `Edit:Add Connection`.

The sections that follow show you how to add network, segment, node, and interface objects to your network map so that they can be managed by NNM.

**Changing Symbol Location on One Submap** After auto-layout is turned off, simply select the symbol and, while holding down the left or center mouse button, drag the symbol to the new location within the submap.

**Copying/Moving Symbols from Submap to Submap** You can easily copy or move symbols from one submap into any other submap in the open map. Select the symbols, then:

- To copy the symbols, select the `Edit:Copy` menu item and `Edit:Paste` menu item. The copy operation does not remove the symbols or selected objects from their original location.
- To move the symbols, select the `Edit:Cut` menu item and `Edit:Paste` menu item.

Sometimes the cut (or delete) operation applies only to a symbol or set of symbols, without affecting the object or objects represented by the symbols. Sometimes the cut operation applies to the object as well as the symbol. When you cut a symbol, if the object that the symbol represents is represented by another symbol, in any map, the symbol is cut, but the object is not deleted from the database. If the symbol is the only symbol that represents the object, the object is cut (or deleted) along with the symbol.

---

**NOTE**

If you are using NNM for UNIX operating systems, you can also move the symbol by dragging and dropping symbols from one submap to another.

---

### Adding Object Symbols to Your Submaps

You can add segment, node, or interface objects to your maps.

**Adding a Segment Object** If NNM can identify which network segment a node is on, it places the node on that segment. If NNM cannot identify the segment, it places the node on the default segment submap for that network. The default segment submap for each network is the first submap created by the `ipmap` service when NNM ran initial discovery (named Segment 1). If this submap has been deleted, the oldest segment submap (lowest segment number) becomes the default segment submap.

---

**NOTE**

You may wish to check the oldest segment submap for each network (Segment 1) periodically to check for stray objects.

---

NNM discovers new nodes on segments attached to connector devices that support the Bridge MIB (RFC1493) and Repeater MIB (RFC2108), as well as HP SNMP bridges and multi-port repeaters (hubs) that have IP or IPX addresses.

A Network submap may contain symbols in the Network and Connector classes. NNM recognizes the following symbols on a Network submap.

**Table 8-1**                      **Symbols Managed on a Network Submap**

<b>Class</b>	<b>Symbol</b>
Network	Bus
Network	Star
Network	Token Ring
Network	FDDI Ring

**Table 8-1**                      **Symbols Managed on a Network Submap (Continued)**

Class	Symbol
Connector	Gateway (Router)
Connector	Bridge
Connector	Repeater
Connector	Multi-Port (Hub)
Connector	Switch (supplied by vendors)

You can add a segment object to a Network submap. When you add a segment, an empty Segment submap is created. Double-clicking on the symbol opens the Segment submap. See NNM's online help for more information.

**Adding a Node Object** You can add an object that represents a node or a network device to your Segment submap by placing one of the supported symbols on a Segment submap. Double-clicking on the node symbol opens a Node submap. NNM manages the following symbols from the Computer, Connector, and Net Device classes in a Node submap. See NNM's online help for more information.

**Table 8-2**                      **Symbols Managed in a Segment Submap**

Class	Symbol
Computer	Generic
Computer	PC
Computer	Workstation
Computer	Mini
Computer	Main Frame
Connector	Bridge
Connector	Gateway (Router)
Connector	Repeater
Connector	Multi-port (Hub)

**Table 8-2** Symbols Managed in a Segment Submap (Continued)

Class	Symbol
Net Device	Analyzer
Net Device	Modem
Net Device	X.25
Net Device	PBX

**Adding an IP or IPX Interface Object** You can add an interface to a Node submap by placing an interface symbol on the Node submap. To do this, enter the address of the interface in the `Add Object` dialog box. See NNM's online help for more information.

**Table 8-3** Symbols Managed in a Node Submap

Class	Symbol
Cards	IP Interface
Cards	IPX Interface

## Background Graphics

A background graphic, such as a map or picture, may be displayed in the background plane of a submap window. The background graphic may be different for each submap. NNM includes an extensive collection of background graphics for your convenience. You can also create your own background graphics.

Keep in mind that graphic files are quite large and may significantly increase the time required to open each submap and significantly increase the amount of memory required to view a submap. If the submap is transient, it becomes persistent when you add a background graphic. Supported file formats for background graphics are as follows:

- *NNM for all platforms:*
  - GIF (CompuServe Graphics Interchange Form)
- *NNM for a Windows or UNIX operating system:* Supports these graphic file formats (not displayed when accessing NNM over the web):

- BMP (Bitmap)
- TIFF (Tag Image File Format)
- *NNM for UNIX operating systems only*: Supports these graphic file formats (not displayed in NNM for Windows or NNM on the web):
  - XBM (X11 monochrome bitmap format)
  - PCX
  - Image
  - Starbase
  - XPM (X Pixmap format)
  - XWD (X Window Dump)
- *NNM for UNIX and NNM on the web only*: Supports these graphic file formats (not displayed in NNM for Windows):
  - JPEG (Joint Photographics Experts Group)

---

**TIP**

The following *analytical data* directories are *not* included in NNM's backup scripts:

- *Windows*:

```
install_dir\backgrounds
install_dir\bitmaps
install_dir\www\htdocs\bitmaps
install_dir\www\registration
```

- *UNIX*:

```
$OV_BACKGROUNDs
$OV_BITMAPs
$OV_WWW/htdocs/bitmaps
$OV_WWW_REG
```

If you make changes or additions to map backgrounds, bitmap files, or web registration files ensure that your new files are properly backed up. See “Backup/Restore to Protect Your Investment of Time” on page 149 for more information.

---

### Adding or Replacing a Background Graphic

You can add a graphic to the background of a submap or replace a graphic with a new graphic. Background graphics shipped with NNM are stored in:

- *Windows:* `install_dir\backgrounds\*.*`
- *UNIX:* `$OV_BACKGROUND`

A graphic can serve as a background for the symbols in a submap. A graphic helps provide contextual information such as a floor plan for systems, a map of geographically diverse sites, and so on. Both users and applications can specify a background graphic for a submap.

The map must be open with read-write access to choose a new background graphic. Select `Map:Submap->Properties` and specify the name of the graphic file that you wish to use.

---

#### NOTE

When using a background graphic, symbols cannot be placed outside the dimensions of the background graphic.

---



## Creating Your Own Map Symbols

You can create your own symbols to use on maps. You might want to do this to customize the appearance of existing symbols, or you might need a symbol to represent a new object. Each symbol is comprised of two graphic elements: the class that determines the background shape of the symbol and changes color according to current status, and the subclass that determines the graphic displayed within the background shape. Several steps are required. See “Changing All the Symbols for a Particular Device” on page 623 for a detailed example of these steps:

- Create a set of \*.GIF images according to the specifications in NNM’s online manual, *Creating and Using Registration Files*.
- Create a symbol registration file to define the characteristics and appearance of the new symbol.
- Develop custom online help information about the your new symbol. This online help information will be available through the symbol’s pop-up menu. See NNM’s online manual, *Creating and Using Registration Files*.

---

### NOTE

These changes are visible from remote consoles and web access after the map is reloaded or refreshed on each system. Your new symbol will also display in `Help:Display Legend`.

---

## What Is a Symbol

A **symbol** is a graphical representation of an object (page 189). A single object can be represented by multiple symbols. Multiple symbols of the same object can exist on the same submap, on multiple submaps of the same map, or on submaps of different maps.

In addition to representing objects, symbols have other functions:

- Most symbols are explodable--when you double-click an explodable symbol, a new submap opens to let you “look inside” the object representing the symbol.

- Some symbols are executable. When you double-click on this type of symbol, a predefined action is carried out on a predefined target.
- Symbols can reflect the status of the object that they represent or of objects in child submaps.

When you create a new symbol type you can define the following characteristics:

Symbol Variety	Symbols come in two varieties: the icon symbol and the connection symbol.				
	<table> <tr> <td>Icon Symbol</td> <td>These symbols represent the objects or groups of objects managed by NNM.</td> </tr> <tr> <td>Connection Symbols</td> <td>These symbols represent logical connections between devices that are managed by NNM. Connection symbols do not represent actual physical connectors.</td> </tr> </table>	Icon Symbol	These symbols represent the objects or groups of objects managed by NNM.	Connection Symbols	These symbols represent logical connections between devices that are managed by NNM. Connection symbols do not represent actual physical connectors.
Icon Symbol	These symbols represent the objects or groups of objects managed by NNM.				
Connection Symbols	These symbols represent logical connections between devices that are managed by NNM. Connection symbols do not represent actual physical connectors.				
Symbol Type	Symbol type consists of the symbol class and subclass (see Note on page 251):				
	<table> <tr> <td>Class</td> <td>You can distinguish the symbol class by the outer shape of the symbol. For example, a circle=a network, square=a computer, diamond=a connective device, and so on.</td> </tr> <tr> <td>Subclass</td> <td>A subclass is a further definition of the symbol that is represented by a bitmap within the outer shape that represents the class.</td> </tr> </table>	Class	You can distinguish the symbol class by the outer shape of the symbol. For example, a circle=a network, square=a computer, diamond=a connective device, and so on.	Subclass	A subclass is a further definition of the symbol that is represented by a bitmap within the outer shape that represents the class.
Class	You can distinguish the symbol class by the outer shape of the symbol. For example, a circle=a network, square=a computer, diamond=a connective device, and so on.				
Subclass	A subclass is a further definition of the symbol that is represented by a bitmap within the outer shape that represents the class.				
Status Source	A symbol may present status information from one of three sources. This characteristic allows applications to have more control over the presentation of status. See “Controlling Symbol Status” on page 256 for more information. In general, Hewlett-Packard recommends that you do not change these settings.				
Label	A label may appear below each symbol. This label is simply a name to help you identify a specific symbol. (see Note on page 251).				

Behavior	Behavior defines how the symbol behaves when you double-click the symbol. A symbol may behave in one of two ways: it may open a child submap (explode), or it may execute an action (execute).
Default Layout	When an explodable symbol opens a new submap, its default layout specifies the layout algorithm (Ring, Bus, Star, Point to Point, or Row/Column) for the submap.

---

**NOTE**

The `ipmap` service automatically assigns a symbol type and label based upon information it receives from the `netmon` service.

This means that if you modify the symbol type or symbol label of an object that is managed by NNM, your changes will be replaced by the values assigned by NNM the next time configuration polling runs.

You can turn off this `ipmap` capability by setting the environment variable `IPMAP_NO_SYMBOL_CHANGES=TRUE`. See the *ipmap* reference page in NNM's online help (or the UNIX manpage) for information about changing this environment variable.

Alternatively, you can modify the appropriate file in the `oid_to_sym_reg` directory to specify that a different symbol be associated with a specific *type* of device (determined by `sysObjectID`). See the *oid\_to\_sym* reference page in NNM's online help (or the UNIX manpage) for more information. See "Changing All the Symbols for a Particular Device" on page 623 for a complete example.

---

## Specifying the Placement and Size of the Submap Windows

You can control the size and placement of the submap windows through the menu choices `Window Geometry` and `Submap Overlay`. The choices you make here affect the way that the submaps are displayed on the management station and on remote consoles, but not through web access.

### Window Geometry

---

**NOTE**

Window geometry settings are visible from remote consoles when the map is reloaded on each system. This change does not impact maps accessed over the web.

---

Select `View:Window Geometry` to explicitly save the size and placement of submap windows so that they are consistent each time they are opened. This feature allows you to override any settings that an application may specify for the size and placement of its submap windows.

When you specify the size or placement of a submap window in a read-write map, those settings will be used in that session and in all subsequent sessions. You cannot change window geometry on read-only maps.

If the submap is transient, it becomes persistent when you save its window geometry.

### Submap Overlay

---

**NOTE**

Submap overlay settings are visible from remote consoles when the map is reloaded on each system. This change does not impact maps accessed over the web.

---

Select `View:Submap Overlay` to minimize the number of windows on your screen during submap navigation. By default, NNM automatically reuses submap windows. Each new submap that you access by double-clicking on it is placed in the window from which you accessed it. This eliminates the display of submaps that you are merely passing through as you navigate the hierarchy. By using submap window overlay, you no longer have to close unneeded submaps that are on your display.

You can change this behavior so that a new window is presented whenever you access a new submap. The overlay feature is set on a per-submap basis. You cannot change submap overlay on read-only maps.

If the submap is transient, it becomes persistent when you turn off overlay.

## Miscellaneous Configuration Changes

A variety of configurable settings are stored in:

- *Windows*: The system registry.
- *UNIX*: The X Windows resource files (page 255).

The changes discussed in this section affect only the local management system's display of NNM or the local remote console's display of NNM. NNM accessed over the web is not affected.

### Windows Operating System

See the *app-defaults* reference page in NNM's online help for detailed information about which NNM settings can be controlled through the Windows system registry.

You can customize the appearance and behavior of NNM by editing the Windows system registry. Check your Windows operating system documentation for more information about editing the system registry files.

Changing the Windows system registry on the management station affects *only* the sessions run on the management station. Each management station and each remote console could have different settings.

Possible settings include:

- Map fonts
- Status colors for symbols and map background color

These settings are stored in the registry under the following keys:

- `\HEY_LOCAL_MACHINE\SOFTWARE\Hewlett-Packard\OpenView\Network Node Manager\OVw\Map Fonts`
- `\HEY_LOCAL_MACHINE\SOFTWARE\Hewlett-Packard\OpenView\Network Node Manager\OVw\Map Colors`

---

**TIP**

After making your changes, check `Help:Display Legend` to verify that your color choices are working successfully.

---

## UNIX Operating Systems

You can also customize the appearance and behavior of NNM by editing the application defaults files and the X resources files used by your X Windows resource manager.

Open the `$APP_DEFS/OVw` file and read the instructions within the file.

Changing the `$APP_DEFS/OVw` file on the management station affects *only* the sessions run on the management station. Each management station and each remote console could have different settings. Each user on the system can have a private X resources file that can override the `$APP_DEFS/OVw` file.

The UNIX `$APP_DEFS/OVw` file for NNM lets you customize things such as:

- Map background color
- Status colors for symbols
- Submap window height and width
- Timers for response to requests to close a map
- Timers for queries
- Keyboard focus policy
- Fonts for dialog boxes

To change general X Windows behavior and color, refer to the online help for the window manager that is being used on your computer. Refer to documentation on your window manager for the location of the X resources file for that window manager.

---

**TIP**

After making status color changes, check `Help:Display Legend` to verify that your color choices are working successfully.

---

## Controlling Symbol Status

You can customize the status color and status propagation of the symbols:

- Control the colors used to indicate the status of network objects.
- Control the way in which critical status is propagated up the map's network hierarchy (“Compound Status” on page 260).
- Monitor the status of router interfaces outside of your firewall (“SNMP Status Polling” on page 259).

## Object Status Colors

NNM displays status conditions via the colors of the symbols. The colors used to display status in NNM's web interface cannot be changed. However the colors used to display status in NNM for a Windows or UNIX operating system can be changed; colors used for both device symbols and connection symbols are defined and documented in:

- *Windows*: The system registry.
- *UNIX*: \$APP\_DEFS/OVw

You can view the current colors using `Help:Display Legend`.

Someone on your team may need to change their colors, for example to address color blindness issues. Keep in mind that your changes affect *only the local display* of the management station or remote console, and must be redone each time you upgrade NNM.

---

### TIP

After making your changes, check `Help:Display Legend` to verify that your color choices are working successfully.

---

To change the colors, see “Miscellaneous Configuration Changes” on page 254.



## Symbol Status

Status is displayed by a symbol on a submap. There are two categories of status conditions: administrative and operational. These two categories are important because they interact with the status propagation schemes in very different ways.

In the **status propagation** schemes, rules are used to determine how the status of a symbol in a child submap is transmitted to the symbols of objects in parent submaps. Administrative status conditions are ignored by the propagation schemes, which means that status information from a child submap is not transferred to a parent symbol. Operational status conditions, on the other hand, may indicate problems that must be propagated through the use of the schemes.

NNM recognizes ten status conditions, shown in the following table:

**Table 8-4** Default Status Conditions

Category	Status Condition	Meaning of Status
Administrative	Unmanaged	Users can set this value, which indicates that the object should not be monitored and that the operational status should be ignored.
Administrative	Testing	An application sets the status to “testing” when an object is undergoing temporary diagnostic or maintenance procedures.
Administrative	Restricted	An application sets the status to “restricted” when an object is functioning normally, but it may not be available to all users.
Administrative	Disabled	An application sets the status to “disabled” when an object is inactive (although there may not be anything necessarily wrong with the object).

**Table 8-4** Default Status Conditions (Continued)

<b>Category</b>	<b>Status Condition</b>	<b>Meaning of Status</b>
Operational	Unknown	An application sets the status to “unknown” when the status of an object cannot be determined.
Operational	Normal	An application sets the status to “normal” when the object is in a normal operational state.
Operational	Warning	An application sets the status to “warning” when an object may face a potential problem.
Operational	Minor/Marginal	An application sets the status to “minor/marginal” when an object has a minor problem; the device itself may continue to operate normally.
Operational	Major	An application sets the status to “major” when an object has serious problems; the device itself probably no longer operates normally.
Operational	Critical	An application sets the status to “critical” when the device represented by the object is not functioning.

The status of a particular symbol represents an application’s interpretation of the status of certain object attributes, or characteristics, as defined in NNM’s object database.

For example, the `ipmap` service determines the status of a node object based upon the operational status of IP and IPX interfaces installed in the node. If all of its IP or IPX interfaces are down, then NNM determines that the node's state is critical. From the perspective of the `ipmap` service, the node is nonfunctional. However, another application may consider the same node functional because that application

monitors a different protocol, and it finds that the interfaces for that protocol are fully functional. Multiple symbols for the node object could be used to represent these status states separately.

### Status Source

Status may be obtained from three possible sources. This feature is mainly used by applications so that they can set the status of specific symbols. In general, Hewlett-Packard recommends that you do not change the status source of a symbol. The status source of a symbol is automatically set depending on the application that manages the object.

The three sources of status are:

**Symbol Status Source** This source of status is used by applications that may want to set status on a particular symbol of the object. This allows other applications to set states for other symbols of the same object.

**Object Status Source** This source of status is used by applications that want to set and display the same status state among all the symbols of a particular object.

**Compound Status Source** NNM determines the status state of the symbol based upon compound status rules. Compound status determines how status propagates from symbols in child submaps to symbols in parent submaps.

The following subsections discuss the significance of each status source.

---

#### NOTE

If you change symbol source, the submap containing the symbol is forced to become persistent.

---

### SNMP Status Polling

In some cases a router may have interfaces to which the NNM management station has no route (for example, interfaces located outside of a firewall). You can obtain status information from the SNMP agent for the device, just as you can for layer-2 interfaces, via SNMP queries for `ifAdminStatus` and `ifOperStatus`.

To configure an IP interface for SNMP status polling, use the following procedure.

1. If the file `netmon.snmpStatus` file does not exist you will need to create it in the location listed below.
  - *Windows:* `install_dir\conf\netmon.snmpStatus`
  - *UNIX:* `$OV_CONF/netmon.snmpStatus`
2. Add a line to the `netmon.snmpStatus` file with the IP wildcards you want to have SNMP status polled.
3. To make your changes take effect, do one of the following.
  - From the command line, type `ovstop -v netmon` then type `ovstart -v netmon`.
  - From the command line, type `xnmpolling -event`.
  - Click **Apply** in the **Options:Network Polling Configuration** dialog box.

See the `netmon` and `netmon.snmpStatus` reference pages in NNM's online help (or the UNIX manpages) for more information.

## Compound Status

NNM uses compound status to propagate the status of a symbol in a low-level submap up to parent submaps, to alert you about a problem. Note that status propagation occurs only when a symbol is given one of the six operational status conditions (unknown, normal, warning, minor/marginal, major, or critical). The administrative status conditions (unmanaged, testing, restricted, and disabled) are not propagated.

You can set compound status in the **Status Propagation** section of the **Map Properties** dialog box.

---

### NOTE

The selections that you make affect remote consoles and the web display after the map is reloaded or refreshed on each system.

---

From the perspective of a compound status scheme, a symbol is either:

- **unknown:** a symbol's current status has not yet been determined by an application.
- **normal:** a symbol is functioning as intended and responding appropriately when contacted by an application.

- **abnormal:** a symbol is given one of the four abnormal status conditions (warning, minor/marginal, major, or critical), and is either potentially or currently having problems.

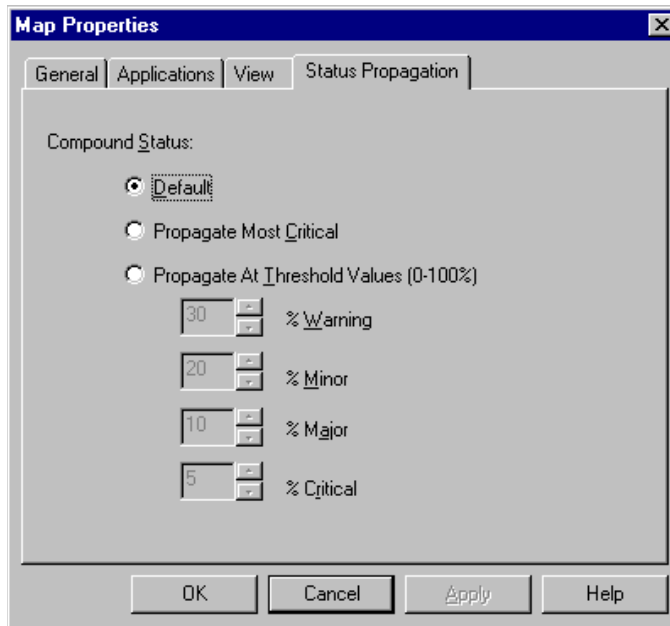
The mapping of the compound status categories to the operational status conditions is described in the following table.

**Table 8-5 Compound Mapping**

This Compound	Maps to This Operational Status Condition
Unknown	Unknown
Normal	Normal
Abnormal	Warning, Minor/Marginal, Major, or Critical

The following subsections discuss each of the three compound status schemes: default, propagate most critical, and propagate at threshold value.

**Figure 8-12 Status Propagation Settings**



### Default

Selecting the default propagation scheme causes NNM to propagate status according to the algorithms described in the following table:

**Table 8-6** Default Status Propagation

Condition of Symbols in Child Submap	Status of Symbols of Parent Object
No symbols are normal and no symbols are abnormal.	Unknown
All symbols are normal.	Normal
One symbol is abnormal; all other symbols are normal.	Warning
More than one symbol is abnormal and more than one other symbol is normal.	Minor/Marginal
One symbol is normal; all other symbols are abnormal.	Major
All symbols are abnormal.	Critical

### Propagate Most Critical

Selecting the propagate most critical scheme causes NNM to propagate the status of the most critical symbol in the child submap to the symbols of the parent object.

### Propagate At Threshold Value (0 - 100%)

Selecting the propagate at threshold value scheme displays four fields that enable you to set threshold values that determine when NNM propagates status. The number shown for each field is its default value.

% <b>warning</b>	30
% <b>minor</b>	20
% <b>major</b>	10
% <b>critical</b>	5

The specified value indicates the minimum percentage of symbols that must be in that status state in order for that status state to be propagated. If the percentage is met or exceeded in more than one status state, then the most severe of those states is propagated.

## Creating New NNM Features to Meet Your Team's Needs

### Adding to NNM's Menu Structure

You can add your own choice of features to NNM's pull-down menus, pop-up menus, and tool bars. This is useful for providing quick access to commonly used tools or to specific network management scripts that your team developed. You may wish to designate a menu location where you post tools developed by team members as a way to share best practices, or designate a menu location to post software tools provided by specific vendors.

There are two ways to make additions to NNM's menu choices:

- NNM provides an easy way to add menu access to MIB-based applications that are written using NNM's MIB Application Builder program. See "Using the MIB Application Builder" on page 425 for more information.
- You can also make modifications to NNM's registration files to add any menu or toolbar items that you want. Complete information about registration files is available in NNM's online manual, *Creating and Using Registration Files*.

---

#### NOTE

The changes that you make using these two options appear on the management station and from remote consoles. They do not show up when NNM is accessed over the web. See Chapter 14, "NNM on the Web," for information about modifying NNM's web user interface.

---

---

#### TIP

You can develop your own *context* settings for your new menu items to make them appear on only specific submaps. See "Using the Context Feature to Control Menu Choices" on page 282. You can also establish *selection rules* that make the menu item unavailable unless an



appropriate device is selected on the map. See NNM's online manual, *Creating and Using Registration Files* for complete information about establishing selection rules.

---

## Creating Executable Symbols with Custom Behaviors

### Symbol Behavior

The setting for a symbol's behavior defines how the symbol behaves when you double-click on it. A symbol may behave in one of two ways. It may explode into a submap, or it may execute an action.

A single object may be represented by both an explodable and an executable symbol. Note that when you change an existing explodable symbol to executable, you can open the child submap only from the `Map: Submap->Open` dialog box, or from another explodable symbol of the parent object.

### Explodable Symbols

If a symbol's behavior is set as explodable (the default), the symbol enables you to view the components of an object. When you double-click on an explodable symbol, NNM opens another submap, called a child submap. The contents of this child submap may be determined by you or an application.

For example, NNM creates a child submap for a node that contains the IP interfaces installed in the node. If you double-click on a computer workstation managed by NNM, a node submap is opened that displays symbols of the node's IP and/or IPX interfaces.

If no child submap exists for the object, when you double-click on the symbol you are prompted to create a child submap. See "Making the Maps Look Like Your World" on page 226 for information about creating child submaps.

### Executable Symbols

You can distinguish executable symbols from explodable symbols by the three dimensional appearance of each executable symbol. A square appears around the symbol indicating that it is executable (button-like). If a symbol's behavior is set as executable, the symbol enables you to

associate the double-click with an action other than opening a child submap. For example, by double-clicking on a conveniently placed symbol, you could:

- Launch a management tool.
- Open a window for remote login on a particular workstation.
- Execute a script that checks the disk space available on a selected system's hard drive and causes a message to display with the results of the check.

You can choose one or more objects as the target objects of the action from among the objects represented on the map. The chosen target objects must be appropriate for the action selected.

---

**TIP**

*NNM over the web*: You will see the executable symbol with a box around it; however, you cannot run the executable.

---

### Creating an Executable Symbol

Using NNM for a Windows or UNIX operating system, you can create a new executable symbol that performs an action. You can copy and paste a symbol, then change the duplicate symbol's behavior so that it executes some action. Or you can change the behavior of an existing symbol from explodable to executable.

When you add an executable symbol to your map, you must select the action that the symbol will execute and the target objects on which it will act. You can choose an action from among the actions supplied:

- `Edit:Add Object` dialog box, choose the `Executable` setting.
- Right-click on a symbol, choose `Symbol Properties`, select the `Executable` setting.

Or you can designate your own action by setting up an application registration file (ARF). The application registration file defines the action and adds it to the list in the `Add Object` dialog box and the `Symbol Properties` dialog box.

For more information about establishing your own custom actions:

- See NNM's online help.
- See NNM's online manual, *Creating and Using Registration Files*.

---

**NOTE**

If an executable symbol is placed on a submap, that submap becomes persistent.

---

Map Customization

**Creating New NNM Features to Meet Your Team's Needs**

---

# **9** **Controlling Map Access**

Customizing the maps requires a substantial time commitment and a level of expertise in the intricate workings of NNM. To protect your efforts, you may want to remove functionality from some maps so that inadvertent detrimental changes cannot be made.

As you are setting up maps for various users/groups, you need to decide how much control you want to maintain over the map. If you allow your team to maintain the map, provide training (see page 272). If you want to maintain control over changes to the map, there are several choices for controlling how NNM looks and functions for each user or group:

- “Setting User Preferences Within NNM” on page 273.

Simply log in as each user/group and set their options. NNM tracks these options by user/group and invokes them each time that user/group accesses NNM. Keep in mind that your team can change these settings if they wish.

- “Using Command Line Startup Options” on page 275.

To use the command line options, you need to develop scripts for each user/group that enforce your choices. Your team member runs the script rather than the regular `ovw` command to open NNM’s interface.

- “Using Operating-System Level File Permissions for the Map” on page 278.

You control access to specific maps through setting operating system level permissions on the `snapinfo.dir` file within each map’s directory.

- “Using the Context Feature to Control Menu Choices” on page 282.

Control which features are accessible through the menus by selecting from predefined “context” identifiers or defining your own. Keep in mind that your team can change these settings if they wish.

- “Using ARF Files to Control Menu Choices” on page 285.

Control which features are accessible through the menus by modifying the underlying application registration files (ARFs). The example provided makes NNM “fool-proof” by removing all access to configuring NNM, yet without limiting the use of NNM. This choice provides you with the ultimate control.

- “Allowing Others to View NNM from Many Computers” on page 303. Learn how to set up multiple systems to run the NNM interface while accessing the common database on the management station. You can set up remote consoles and/or web access.

## **Establishing and Communicating a Process for Requesting Changes to the Map**

Now that your maps are set up just the way you want them, think about how you will handle ongoing changes to the maps.

### **Not Allowing Team Members to Make Changes**

Provide procedures for your team members to follow when requesting changes to their maps. Who should your team members contact and how. Designate one person to maintain the maps.

### **Allowing Team Members to Make Changes**

If you are allowing your team members to have access to editing features, think about what training they will need in order to understand how NNM works and make wise decisions as they use the map. For example:

- All maps use the same database information. Don't delete symbols from the map. Use the hide feature.
- If more than one person uses a particular map, only the first person to open the map will have read/write permissions. Identify which team member has read-write permission for each map.
- If you are viewing a map in read-only mode, use `Map:Refresh` to display the most recent changes.
- Explain what to do when symbols show up in the `New Object Holding Area`. Explain why this might happen, such as when a new device added to your network.
- If they are allowed to modify their submaps, provide information about how to edit maps so that the connections between devices are maintained, and/or how to reestablish connections that are lost from the map.



## Setting User Preferences Within NNM

You can define the following user preferences for each user/group. Simply log into the system as each user/group and set the preferences. NNM keeps track of each user's settings. These settings can be changed by each team member. You may want to simply provide training about these choices and leave it up to each team member, or you may wish to set them up for each team member and then remove the menu choices that allow changes (see page 285):

- Set the default map.
- Set a home submap.
- Set the toolbar to be hidden or displayed.
- Create a list of frequently accessed submaps by adding symbols to the Quick Navigator (this is customized per map rather than per user).

**Table 9-1**                      **User Preference Settings**

<b>User Preference</b>	<b>Setting</b>
The <b>home</b> submap is the first submap that appears when you open a map. The home submap is always persistent.	Define the home submap: <ol style="list-style-type: none"> <li>1. Open the submap you want to be displayed each time you open the current map.</li> <li>2. Select Map:Submap-&gt;Set This Submap as Home</li> </ol> (Not available in NNM on the web.)
The map that NNM first displays upon startup is called the <b>user default map</b> .	Define the user's default map: <ol style="list-style-type: none"> <li>1. Select Map:Open.</li> <li>2. Select a map name.</li> <li>3. Click Set User Default.</li> </ol>
The toolbar provides quick access to frequently-used menu items.	<i>UNIX only:</i> Display or hide the tool bar by selecting View:Toolbar

**Table 9-1** User Preference Settings (Continued)

<b>User Preference</b>	<b>Setting</b>
The Quick Navigator provides quick access to frequently used submaps.	<ul style="list-style-type: none"><li>• Add a submap symbol to the Quick Navigator:<ol style="list-style-type: none"><li>1. Open the submap you want to access from the Quick Navigator.</li><li>2. Select Edit:Add to Quick Navigator-&gt;Add This Submap.</li></ol></li><li>• Remove a submap symbol from the Quick Navigator:<ol style="list-style-type: none"><li>1. Select the symbol in the Quick Navigator and display the symbol pop-up menu.</li><li>2. Select Delete Symbol from the pop-up menu.</li></ol></li></ul>

Remember that your team members can change these settings at any time unless you remove the menu choices for making those changes (see page 285).

## Using Command Line Startup Options

This section discusses details about the `ovw` command that you may use to start NNM from the command line prompt. You can write scripts for each user/group that specify their specific startup options.

### TIP

It may be useful to enter these into each user's login script. For security purposes, remove the *write* permissions to the login scripts after you are done.

These commands may be useful for customizing NNM per user group or useful when running multiple sessions of NNM at the same time.

### Options

The `ovw` command contains the following options. See the *ovw* reference page in NNM's online help (or the UNIX manpage) for more information.

```
ovw [-ro|-rw] [-map map_name] [-submap submap_name] [-propagation
scheme] [-server] [-verify] [-fields] [-copyMap source_map_name
destination_map_name ...] [deleteMap map_name] [mapcount]
```

**Table 9-2** Startup Options for NNM

Option	Meaning
<code>ovw -ro</code>	Open all maps as read-only.
<code>ovw -rw</code>	Open all maps as read-write if possible. <i>UNIX</i> : This is the default. It is necessary only if the X resources specify read-only as a default behavior.
<code>ovw -map map_name</code>	Open map <i>map_name</i> . Create the map, <i>map_name</i> , if it does not exist.
<code>ovw -submap submap_name</code>	Open <i>submap_name</i> as the initial submap if it exists on the map being opened. If <i>submap_name</i> does not exist, then NNM initially opens the home submap of the map being opened.

**Table 9-2 Startup Options for NNM (Continued)**

Option	Meaning
ovw <i>-propagation scheme</i>	Sets the propagation scheme for a new map or changes the scheme for an existing map. The propagation schemes are default, most critical, and threshold.

### Examples

Table 9-3 provides examples that show you various ways you can use the `ovw` command to start NNM from a command line prompt.

**Table 9-3 Using the `ovw` Command to Start NNM**

Command Entered	Effects
ovw	Runs NNM and opens the map assigned as the user default map.
ovw <code>-map Admin_Map</code>	Runs NNM and opens the map named <code>Admin_Map</code> . <code>Admin_Map</code> does not need to be the user default map.
ovw <code>-map world</code> <code>-propagation</code> <code>threshold:40:30:20:5</code>	Runs NNM and creates a new map named <code>world</code> using the threshold propagation scheme with a warning threshold of 40%, a minor threshold of 30%, a major threshold of 20%, and a critical threshold of 5%.
ovw <code>-ro</code>	Runs NNM and opens the map assigned as the user default map. The user default map, and all maps subsequently opened during this session, are opened with read-only access.
ovw <code>-ro -map Europe2</code>	Runs NNM and opens the map called <code>Europe2</code> . All maps during this session of NNM are opened with read-only access.
ovw <code>-rw</code>	Runs NNM and opens the map assigned as the user default map. Maps can be opened with read-write access if permissions allow. The <code>-rw</code> option is the default and is necessary only if read-only is specified as the default behavior.

**Table 9-3** Using the `ovw` Command to Start NNM (Continued)

<b>Command Entered</b>	<b>Effects</b>
<code>ovw -map Admin_Map -submap Main</code>	Runs NNM and opens map named <code>Admin_Map</code> , and first displays the submap named <code>Main</code> instead of the home submap.

## Using Operating-System Level File Permissions for the Map

Two factors determine whether a map is opened with read-write access or with read-only access when a map is opened. First, only one user can have a specific map opened with read-write access at any one time (the first person to access the map). Second, you can use the file system to purposely allow specific users read-write or read-only access to a map by setting permission to the files.

---

**TIP**

If you want several users to have simultaneous read-write access, make copies of the map for each user. See “Copying the Default Map” on page 201. Keep in mind that if your team members make changes to their map, it will then be out of sync with the other maps. To prevent team members from making changes to the map, see page 285.

---

Possible permission settings are:

**Read-write Access** A map opened with read-write access is completely editable. You can add objects, add connections, create submaps, and change object attribute values. Only one user can have a specific map open with read-write access at any one time. If another user has the map open with read-write access, when you open the same map, you have read-only access to it. This is the case even if you have read-write permission from the operating system.

**Read-only Access** A map opened with read-only access is not editable by the user, although the map still can receive status from applications. In a map open with read-only access, you can view status changes, perform locate operations on objects, and update topological changes using the `Map:Refresh Map` menu item. However, you cannot add, delete, or modify items in the map, including symbols, objects, submaps, and map snapshots.

**CAUTION:** Using `Map:Open` or `Map:Save as`, it may be possible to navigate to a read-write map.

You can use command line options to make *all* maps read-only (page 275) or remove these menu choices (page 285).

No Access      You cannot open a map that is set with No Access.

## Windows: Setting Permissions

User access to maps is governed by the file permissions of the `snapinfo.dir` file in each individual map directory. If NNM is installed on:

- *FAT file system*, you can make each map read-only or read-write for *everyone* on the system. HP recommends that you install NNM on an NTFS file system if more individualized map access permissions are a concern.
- *NTFS*, you have a fine level of control over access to maps. For example, using the Windows Explorer's *Security* tab, you can adjust the permissions so that specific users/groups have write access to specific maps only.

### FAT file systems

---

#### CAUTION

Close the NNM interface before changing map permissions (see “Closing All Current Sessions” on page 305). NNM checks permissions only when it opens the map. If, for example, you change permissions for an open map from read-write to read-only, you risk map database corruption.

---

Each NNM map is comprised of a group of files. NNM needs to be able to make continuous changes to these files. A special file that controls users' access to the map files themselves is provided for each map. You can set individual maps to read-write or read-only by setting the attributes on the following file:

```
install_dir\databases\openview\mapdb\mapname\snapinfo.dir
```

Log in as Administrator. Use the Windows Explorer to access the file properties of the `snapinfo.dir` file. When you set this file to read-only, all the files that comprise the map are read-only protected for *all* users. The next time the map is opened, your changes take affect.

## NTFS

---

### CAUTION

Close the NNM interface before changing map permissions (see “Closing All Current Sessions” on page 305). NNM checks permissions only when it opens the map. If, for example, you change permissions for an open map from read-write to read-only, you risk map database corruption.

---

Each NNM map is comprised of a group of files. A special file that controls users’ access to the map files themselves is provided for each map. You can set customized permissions for each user/group on individual maps. Set read-write or read-only attributes on the following file:

```
install_dir\databases\openview\mapdb\mapname\snapinfo.dir
```

Log in as Administrator. Access the file properties of the `snapinfo.dir` file and use the Windows Explorer’s File:Properties-> Security tab. Set the appropriate permissions for each user or group that accesses this file. When you set this file to read-only, all the files that comprise the map are read-only protected for *the specific user/group*. The next time the map is opened, your changes take affect.

## UNIX: Setting Permissions

---

### CAUTION

Close the NNM interface before changing map permissions (see “Closing All Current Sessions” on page 305). NNM checks permissions only when it opens the map. If, for example, you change permissions for an open map from read-write to read-only, you risk map database corruption.

---

Each NNM map is comprised of a group of files. A special file that controls user’s access to the map files themselves is provided for each map. You can set customized permissions for each user/group on individual maps. Set read-write or read-only attributes on the following file:

```
$OV_DB/Openview/mapdb/mapname/snapinfo.dir
```



To change map permissions, log in as root. Use the commands provided by NNM. *Do not use the UNIX file permission commands (chmod, chgrp, chown) from your operating system.*

NNM provides several commands to let you change permissions:

- |          |   |
|----------|---|
| ovwchmod | Changes the mode of individual maps.                    |
| ovwchgrp | Changes the group ownership of individual maps.         |
| ovwchown | Changes the ownership of individual maps.               |
| ovwperms | Lists the available maps and their current permissions. |

See the *ovwperms* reference page in NNM's online help (or the UNIX manpage) for more information.

To change permissions of *other* NNM configuration files and database files, use the UNIX *chown*, *chmod*, and *chgrp* commands. You may need to stop and start NNM's services (background processes) in order to force NNM to acknowledge the changes. You do not need to stop and start any services to force NNM to acknowledge the change in *map* permissions. The next time the map is opened, your changes take affect.

## Using the Context Feature to Control Menu Choices

---

### NOTE

This feature is not available when accessing NNM over the web.

If you want to remove completely a menu choice from all submaps rather than from specific submaps, see page 285.

If you want to add menu choices, the context feature allows you to control in which submaps your new choices display. See “Creating New NNM Features to Meet Your Team’s Needs” on page 264, and see also “Using the MIB Application Builder” on page 425.

---

This feature allows you to remove or display specific *groups* of pull-down menu items and pop-up menu items on a submap-by-submap basis.

There are two components to the context feature:

- Application Registrations Files (ARFs)  
Menu items are defined in application registration files. Each menu item is set to display in one or more *contexts*.
- Map:Submap->Properties / Context tab  
Each submap is configured to display one or more *contexts*.

NNM provides multiple predefined *contexts* for your use. You can also create your own *contexts*.

Anyone with read-write access to the map can change the contexts assigned to a submap.

### Using NNM’s Predefined Contexts

NNM’s predefined context identifiers that are provided for your convenience are:

- `isInternet`  
Displays menus, toolbar buttons, and commands pertaining to an Internet network.

- `isIP`  
Displays menus, toolbar buttons, and commands pertaining to an Internet Protocol (IP) network.
- `isIPX`  
Displays menus, toolbar buttons, and commands pertaining to an Internet Packet Exchange (IPX) network.
- `isNetwork`  
Displays menus, toolbar buttons, and commands pertaining to network management.
- `isNode`  
Displays menus, toolbar buttons, and commands pertaining to node management.

You can add and remove these predefined context identifiers through the **Context** tab in the `Map:Submap Properties` dialog box or the **New Submap Wizard**. See NNM's online help from these dialog boxes for more information.

---

**NOTE**

If you see the `NoGeneric` or `NoDefault` context identifier in a submap's properties, do not change it. These are special contexts utilized by application developers.

---

## Creating Your Own Contexts

Contexts are specified on a menu-item by menu-item basis in the application registration files (ARFs) that control HP OpenView or third-party applications. (See page 291 for an example ARF file.)

To create your own context IDs:

- See NNM's online manual, *Creating and Using Registration Files*, for details about editing application registration files (ARFs).
- See the *OVwRegIntro* reference page in NNM's online help (or the UNIX manpage) for more information.

---

**NOTE**

If you don't want a menu item to show up everywhere, delete the `AllContexts` identifier associated with that menu item in the ARF file, and replace it with a more specific context identifier.

---

Close the NNM user interface (`ovw` service) and reopen it to force NNM to acknowledge changes to the ARF files.

---

## Using ARF Files to Control Menu Choices

---

### NOTE

These changes do not affect NNM over the web. NNM's web-based user interface is read-only for all maps, so you do not need to do anything to prevent team members who have web access from making changes to the map.

---

You can configure NNM so that your team cannot accidentally change your map customizations while accessing maps from remote consoles or on the management station itself. In this example, all features that allow access to modifying NNM are removed *without* setting the map to read-only. By leaving the read-write access to the map, your team retains full use of NNM and can still make map snapshots as needed. You are left in complete control over how NNM operates. Your team can use NNM, but cannot change any configuration settings. All access to configuration settings is removed from your team's view of the software, but remains in your view of the software.

See NNM's online manual, *Creating and Using Registration Files*, for details about editing application registration files (ARFs).

1. Establish *login accounts* for each user/group on your team. NNM needs to be able to differentiate between them in order to provide customized menu selections.
2. Create a unique directory for each user/group to hold their custom registration files. For example:

- *Windows*: You should be the only user with read-write access:

```
install_dir\registration\connectivity_reg\C\*. *  
install_dir\registration\printers_reg\C\*. *  
install_dir\registration\servers_reg\C\*. *
```

- *UNIX*:

```
/etc/opt/OV/share/registration/connectivity_reg/C/*  
/etc/opt/OV/share/registration/printers_reg/C/*  
/etc/opt/OV/share/registration/servers_reg/C/*
```

(Set your new directories and the files that you place within them to have at least permissions of r-xr-xr-x.)

---

**TIP**

---

Placing your custom registration files in this location ensures that your work is included in the NNM backup and restore process.

3. Copy NNM's default registration directory to the new empty directories. NEVER MODIFY THE ORIGINAL FILES:

- *Windows*: in Windows Explorer, select the contents of `install_dir\registration\C\*.*` and paste it into each new directory that you created in the previous step.
- *UNIX*: repeat the following command for each new directory that you created in the previous step.

```
# cd $OV_REGISTRATION/C
# find . | cpio -pdumv $OV_REGISTRATION/each new directory/C/
```

4. Delete the following files from each new directory that you just populated:

- `dataWarehouse`  
To disable access to NNM's data warehouse feature that exports the NNM database contents to a relational database.
- `xnmpolling`  
To disable access to NNM's polling configuration settings.
- `ovsnmp/xnmbuilder`  
To disable access to NNM's Application Builder program.
- `ovsnmp/xnmcollect`  
To disable access to setting up or changing data collection and threshold monitoring.
- `ovsnmp/xnmloadmib`  
To disable access to changing the SNMP MIBs loaded on your system.
- `ovsnmp/xnmsnmpconf`  
To disable access to customizing NNM's record of your network's SNMP configuration settings such as GET- and SET-community names and port settings.

- `ovsnmp/xnmtrap`

To disable access to customizing NNM's event configuration settings.

5. In each of the new directories that you just created, open the following application registration file (ARF): `...\\C\\ovw`

Use an ASCII editor program and comment out or delete the following blocks of code that provide potentially destructive functionality to the user interface. To comment out a section, use the `/*` characters at the beginning and the `*/` characters at the end of the section that you wish NNM to ignore. Although you can delete these sections, for the convenience of troubleshooting and changing your mind later, it is better to simply comment them out.

---

**TIP**

---

Never place comments within comments.

The ARF file controls which menu choices show up in the NNM interface. If you find it confusing to read the file, refer to the NNM user interface for clues about what you are seeing. Refer to the *OVwRegIntro* reference page in NNM's online help (or the UNIX manpage) for information about the syntax of the file.

- In the MenuBar `<100>` "Map" `_M` section, delete or comment out the following sections to remove Map pull-down menu items from the user interface (see the example ARF file on page 291):

```
"New..."
"Open..."
"Save As..."
"Export..."
"Import..."
"Delete..."
"Properties..."
"Submap..."

Menu Submap
"New..."
"Open..."
"Delete..."
"Delete This Submap"
"Set This Submap As Home"
```

```
"Make This Submap Persistent"  
"Properties..."  
"Close"
```

- In the MenuBar <100> "Edit" \_E section, delete or comment out the following sections to remove Edit pull-down menu items from the user interface (see the example ARF file on page 291):

```
"Add Object..."  
"Add Connection..."  
"Cut"  
"Copy"  
"Paste"  
"Delete"  
"Manage Objects"  
"Unmanage Objects"  
"Object Properties..."
```

```
Menu Delete  
"From This Submap"  
"From All Submaps"
```

- In the MenuBar <100> "View" \_V section, delete or comment out the following sections to remove View pull-down menu items from the user interface (see the example ARF file on page 291):

```
"Hidden Objects"  
"Automatic Layout"  
"Redo Layout"
```

```
Menu "Hidden Objects"  
"Hide Selected From This Submap"  
"Hide Selected From All Submaps"  
"Show Hidden On This Submap"  
"Show Hidden On All Submaps"
```

- In the Menu Find section, delete or comment out the following sections to remove Find menu items from the user interface (see the example ARF file on page 291):

```
"Submap..."
```

- Delete or comment out all of the following to disable access to NNM customization through the pop-up menus (see the example ARF file on page 291):



```
"Change Symbol Type"  
"Symbol Properties"  
"Delete"  
"Hide"  
"Set Star Center"
```

6. Verify that your changes are valid. At the command prompt, type:

```
ovw -verify
```

7. In each of the new directories that you just created, open the following application registration file (ARF): ...\\C\\ipmap

Use an ASCII editor program and comment out or delete the following blocks of code that provide potentially destructive functionality to the user interface. To comment out a section, use the `/*` characters at the beginning and the `*/` characters at the end of the section that you wish NNM to ignore. Although you can delete these sections, for the convenience of troubleshooting and changing your mind later, it is better to simply comment them out.

---

**TIP**

---

Never place comments within comments.

The ARF file controls which menu choices show up in the NNM interface. If you find it confusing to read the file, refer to the *OVwRegIntro* reference page in NNM's online help (or the UNIX manpage) for information about the syntax of the file.

In the MenuBar "Map" section, delete or comment out the following sections to remove pull-down menu items from the user interface:

```
"Import..."  
"Export..."
```

8. Verify that your changes are valid. At the command prompt, type:

```
ovw -verify
```

9. Set the `ovw` environment variable `OVwRegDir` in each user/group's path so that the new registration files are used each time this user logs in:

- *Windows*: Edit the system registry files:
  - a. Select `Start:Run` and type **regedit**.

- b. Add a new string value to HKEY\_USERS\*S-id#*\Environment.  
For example:

Name: **OVWRegDir**

Data: **c:install\_dir\registration\printers\_reg\C\**

---

**NOTE**

If you are unsure which *S-id#* (systemID number) is the correct user *id#*, you can reference the registry entry and check the ProfileImagePath data. Look at the last 3 or 4 digits in the *S-id#*:

HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\WindowsNT\current version\ProfileList\*S-id#*.

- 
- **UNIX:**  
C shell users, enter the following in each `.cshrc` file; for example:

```
setenv OVwRegDir=/etc/opt/OV/share/registration/printers_reg/C/
```

Borne and Korn shell users, enter the following in each `.profile` file; for example:

```
export OVwRegDir=/etc/opt/OV/share/registration/printer_reg/C/
```

10. Log in as the user you just established, and test the look and feel of the user interface to make sure you obtained the results that you had in mind. Try out several submaps with a variety of context settings.

---

**TIP**

If it isn't working as you expected, you can always return to the untouched original files. In a Windows operating system, however, the registry entries will not be automatically reset.

11. When everything is working as you planned, set the file permissions on your custom registration files as follows:
  - **FAT file system in Windows**, you can set individual files to read-only for *everyone* on the system. Log in as Administrator. Use the Windows Explorer to access the file properties of each file in the custom registration directory. Set these files to read-only to prevent all users from changing your customizations.

---

**TIP**

As a workaround for the limited user-level permissions in FAT, consider creating an NTFS partition on your hard drive. Then use the partitioned space for your customized registration directory in order to gain more control over access security. However, if you do this, your customized registration files are no longer automatically included in NNM's backup routine.

- *NTFS in Windows*, you have a fine level of control over file permissions. Log in as Administrator. Using Windows Explorer's File:Properties->Security tab, set up the appropriate permissions for the *directory* that holds your customized registration files. Be sure to select both Replace Permissions on Subdirectories and Replace Permissions on Existing Files. Set the permissions to read-only for each user/group to prevent them from changing your customizations.
- *UNIX*: set the appropriate permissions on each file using the standard UNIX `chmod` and `chown` commands. For example:

```
# chmod -R 500 /etc/opt/OV/share/registration/printers_reg/C
# chown -R printers /etc/opt/OV/share/registration/printers_reg/C
# chgrp -R bin /etc/opt/OV/share/registration/printers_reg/C
```

## ARF File Modifications Example

The following is a printout of the `ovw` ARF file with bold type showing the sections discussed in the previous section. Place comment symbols `/* */` around all the sections in bold type to implement the suggested changes on your management station.

---

**TIP**

Do not place comments within comments.

```
/*
**
** @(#)HP OpenView Windows Menu Registration
**
** $Revision: /main/PULSAR/5 $
** $Date: 1998/07/06 18:43 UTC $
```

## Controlling Map Access

### Using ARF Files to Control Menu Choices

```
*/
Application "OpenView Windows"
{
    Description {
        "HP OpenView"

        "Application registration fiel for various menu items",

        "which present themselves from the ovw menu bar."

    Version "Network Node Manager X.X (copy_proot modifies)";
    Copyright {
        "Copyright (c) 1990-19XX Hewlett-Packard Company (copy_proot modifies)",
        "All rights reserved."
    }
}
#endif NT
    HelpDirectory "OVW";
#endif /* NT */

/*
**
** OVw Menu Bars
**
*/
MenuBar <100> "Map" _M
{
#ifdef SPECIAL_EDITION

    <100>  "New..."          _N      Ctrl<Key>N
        Context (AllContexts || WantMapMenus)
        f.new_map;
    <100>  "Open..."        _O      Ctrl<Key>O
        Context (AllContexts || WantMapMenus)
        f.avail_maps;
    <100>  "Refresh"         _f
        Context (AllContexts || WantMapMenus)
        f.refresh_map;
    <100>  "Save As..."     _A      Ctrl<Key>S
        Context (AllContexts || WantMapMenus)
        f.save_map;

    /* 99 is used for ipmap and map customization */

    <98>   "Delete..."      _D
        Context (AllContexts || WantMapMenus)
```

```

        f.avail_maps;
#endif /* !SPECIAL_EDITION */
    <98>    "Properties..."    _r
           Context (AllContexts || WantMapMenus)
           f.map_desc;
    <98>    "Separator 1"
           Context (CantTouchThis)
           f.separator;
#endif NT
    <98>    "Print Window"    _P
           Context (AllContexts || WantMapMenus)
           f.action Print_Window;
#endif /* !NT */
    <98>    "Submap"
           _S
           Context (AllContexts || WantMapMenus)
           f.menu "Submap";
#endif SPECIAL_EDITION
    <98>    "Snapshot"
           _h
           Context (AllContexts || WantMapMenus)
           f.menu "Snapshot";
#endif /* !SPECIAL_EDITION */
    <0>     "Exit"
           _x
           Context (CantTouchThis)
           f.exit;
}
#endif NT
Action "Print_Window" {
    Command "printwindow";
}
#endif /* NT */
Menu Submap
{
    <100>    "New..."
           _N
           Context (AllContexts || WantMapMenus)
           f.new_submap;
    <100>    "Open..."
           _O
           Context (AllContexts || WantMapMenus)
           f.list_submaps;
    <100>    "Delete..."
           _D
           Context (AllContexts || WantMapMenus)
           f.list_submaps;
    <100>    "Delete This Submap"
           _T
           Context (AllContexts || WantMapMenus)
           f.delete_smap;
    <100>    "Set This Submap As Home"
           _H
           Context (AllContexts || WantMapMenus)

```

## Controlling Map Access

### Using ARF Files to Control Menu Choices

```
                f.set_home_submap;
#ifdef NT
    <100>  "Make This Submap Persistent"  _M
          Context (AllContexts || WantMapMenus)
          f.persistify;
#endif /* !NT */
    <100>  "Properties..."              _r
          Context (AllContexts || WantMapMenus)
          f.submap_desc;
    <1>    "Close"                        _C      <Key>Escape
          Context (AllContexts || WantMapMenus)
          f.close_smap;
}
#ifdef SPECIAL_EDITION
Menu "Snapshot"
{
    <100>  "New..."                     _N
          Context (AllContexts || WantMapMenus)
          f.create_snap;
    <100>  "Open..."                     _O
          Context (AllContexts || WantMapMenus)
          f.avail_snaps;
    <100>  "Delete..."                   _D
          Context (AllContexts || WantMapMenus)
          f.avail_snaps;
    <100>  "Properties..."               _r
          Context (AllContexts || WantMapMenus)
          f.snap_desc;
    <100>  "Close"                        _C
          Context (AllContexts || WantMapMenus)
          f.close_snapshot;
}
#endif /* !SPECIAL_EDITION */

MenuBar <100> "Edit" _E
{
    <100>  "Add Object..."               _A
          Context (AllContexts || WantEditMenus)
          f.add_obj;
    <100>  "Add Connection..."           _n
          Context (AllContexts || WantEditMenus || WantEditObjs)
          f.add_conn;
    <100>  "Separator 1"
          Context (CantTouchThis)
          f.separator;
}
```

```

<100>  "Cut"                _t      Ctrl<Key>X
        Context (AllContexts || WantEditMenus || WantEditCut)
        f.cut_obj_smap;
<100>  "Copy"               _C      Ctrl<Key>C
        Context (AllContexts || WantEditMenus || WantEditCut)
        f.copy_obj_smap;
<100>  "Paste"              _P      Ctrl<Key>V
        Context (AllContexts || WantEditMenus || WantEditCut)
        f.paste;
<100>  "Delete"             _D
        Context (AllContexts || WantEditMenus || WantEditObjs)
        f.menu Delete;
<100>  "Selected Objects List..." _L      Ctrl<Key>L
        Context (AllContexts || WantLocateMenus)
        f.sel_objs;
<100>  "Find"                _F
        Context (AllContexts || WantEditMenus || WantLocateMenus)
        f.menu Find;
<50>   "Separator 2"
        Context (CantTouchThis)
        f.separator;
<50>   "Manage Objects"     _M
        Context (AllContexts || WantMapMenus)
        f.manage_objects;
<50>   "Unmanage Objects"   _U
        Context (AllContexts || WantMapMenus)
        f.unmanage_objects;
<1>    "Object Properties..." _r
        Context (AllContexts || WantEditMenus || WantEditObjs)
        f.obj_desc;
}

Menu Delete
{
  <100>  "From This Submap"   _T
        Context (AllContexts || WantEditMenus || WantEditObjs)
        f.delete_obj_smap;
  <100>  "From All Submaps"   _A
        Context (AllContexts || WantEditMenus || WantEditObjs)
        f.delete_obj;
}
Menu Find
{
  <100>  "Submap..."        _S
        Context (AllContexts || WantLocateMenus)
        f.list_submaps;
}

```

```

<100>  "Object By Selection Name..."  _N      Ctrl<Key>F
      Context (AllContexts || WantLocateMenus)
      f.locate_name;
<100>  "Object By Attribute..."        _A      Ctrl<Key>A
      Context (AllContexts || WantLocateMenus)
      f.locate_attr;
<100>  "Object By Comment..."         _C
      Context (AllContexts || WantLocateMenus)
      f.locate_comment;
<100>  "Object By Status..."          _u
      Context (AllContexts || WantLocateMenus)
      f.locate_status;
<100>  "Object By Symbol Type..."     _T
      Context (AllContexts || WantLocateMenus)
      f.locate_type;
<100>  "Object By Label..."           _L
      Context (AllContexts || WantLocateMenus)
      f.locate_label;
}
MenuBar <100> "View" _V
{
  <100>  "Highlights"                  _H
      Context (AllContexts || WantEditMenus )
      f.menu Highlights;
  <100>  "Hidden Objects"                _n
      Context (AllContexts || WantEditMenus || WantEditHide)
      f.menu "Hidden Objects";
#ifdef NT
  <100>  "Toolbar"                    _T
      Context (AllContexts || WantViewMenus)
      f.submap_toolbar_toggle;
  <100>  "Status Bar"                 _S
      Context (AllContexts || WantViewMenus)
      f.submap_status_toggle;
#else /* NT */
  <100>  "Toolbar"                    _T
      Context (AllContexts || WantViewMenus)
      f.toolbar_toggle;
#endif /* NT */
  <100>  "Pan and Zoom"                _Z
      Context (AllContexts || WantViewMenus)
      f.show_panner;
#ifdef NT
  <100>  "User Plane"                 _U
      Context (AllContexts || WantViewMenus)
      f.user_plane;

```



```

<100>  "Automatic Layout"      _A
        Context (AllContexts || WantViewMenus)
        f.auto_layout;
<100>  "Submap Overlay"       _O
        Context (AllContexts || WantViewMenus)
        f.overlay_toggle;
#else /* NT */
<100>  "Submap Overlay"       _O
        Context (AllContexts || WantViewMenus)
        f.nt_overlay_toggle;
#endif /* NT */
<100>  "Window Geometry"     _G
        Context (AllContexts || WantViewMenus)
        f.menu "Window Geometry";
<100>  "Redo Layout"         _R
        Context (AllContexts || WantViewMenus)
        f.redolayout;
}
Menu Highlights
{
<100>  "Select Highlighted"   _S  Ctrl<Key>H
        Context (AllContexts || WantEditMenus)
        f.select_highlighted;
<100>  "Clear Highlighted"   _C
        Context (AllContexts || WantEditMenus)
        f.clear_highlights;
}
Menu "Hidden Objects"
{
<100>  "Hide Selected From This Submap"  _T
        Context (AllContexts || WantEditMenus || WantEditHide)
        f.hide_obj_smap;
<100>  "Hide Selected From All Submaps"  _A
        Context (AllContexts || WantEditMenus || WantEditHide)
        f.hide_obj;
<100>  "Show Hidden On This Submap"     _S
        Context (AllContexts || WantEditMenus || WantEditHide)
        f.unhide;
<100>  "Show Hidden On All Submaps"     _O
        Context (AllContexts || WantEditMenus || WantEditHide)
        f.unhide_all;
}

Menu "Window Geometry"
{
<100>  "Save For This Submap"  _T

```

```

        Context (AllContexts || WantViewMenus)
        f.geometry_smap;
    <100> "Save For All Open Submaps"      _A
        Context (AllContexts || WantViewMenus)
        f.geometry_all;
    <100> "Clear For This Submap" _S
        Context (AllContexts || WantViewMenus)
        f.geometry_smap_off;
    <100> "Clear For All Open Submaps"    _O
        Context (AllContexts || WantViewMenus)
        f.geometry_all_off;
    }
MenuBar <100> "Performance" _P
{
}
MenuBar <100> "Configuration" _C
{
}

MenuBar <100> "Fault" _F
{
}
MenuBar <100> "Accounting" _A
{
}
MenuBar <100> "Security" _S
{
}
MenuBar <100> "Tools" _T
{
}
MenuBar <100> "Options" _O
{
}
MenuBar <100> "Window" _W
{
    <100> "Separator 1"
        Context (CantTouchThis)
        f.separator;
#ifdef NT
    <100> "Cascade Submaps"                _C
        Context (AllContexts || WantLocateMenus)
        f.cascade_submaps;
    <100> "Separator 2"
        Context (CantTouchThis)
        f.separator;
#endif
}

```

```

#endif /* NT */
    <100>    "Root"
            Context (AllContexts || WantLocateMenus)
                f.goto_root;
    <100>    "Home"
            Context (AllContexts || WantLocateMenus)
                f.goto_home;
    <100>    "Parent"
            Context (AllContexts || WantLocateMenus)
                f.goto_parent;
    <51>    "Separator 3"
            Context (CantTouchThis)
                f.separator;
}
MenuBar <0> "Help" _H
{
    <90>    "Display Legend"
            Context (AllContexts || WantHelpMenus)
                f.disp_legend;
#ifdef NT
    <80>    "Mouse and Keyboard"
            Context (AllContexts || WantHelpMenus)
                f.help_browser "ovw:mousenkeyboard";
    <80>    "On Window"
            Context (AllContexts || WantHelpMenus)
                f.help_browser "ovw:overv";
    <60>    "Using Help"
            Context (AllContexts || WantHelpMenus)
                f.on_help;
    <50>    "Misc"
            Context (AllContexts || WantHelpMenus)
                f.menu "Misc Help";
#endif /* !NT */

    <43>    "Release Notes" _L f.action "ReleaseNotes";

    <40>    "About HP OpenView..."
            Context (AllContexts || WantHelpMenus)
                f.about_ovw;
}
Action "ReleaseNotes"
{
    Command "ovweb -ReleaseNotes README.html";
}
#ifdef NT
Menu "Misc Help"

```

## Controlling Map Access

### Using ARF Files to Control Menu Choices

```
{
  <98>   "Tasks"           _T
        Context (AllContexts || WantHelpMenus)
            f.task_index;
  <98>   "Functions"       _F
        Context (AllContexts || WantHelpMenus)
            f.function_index;
}
#endif /* !NT */

/*
 * OVw Popup menu
 */
PopupItem <100> "Open" Context (AllContexts || WantPopupMenus)
    TargetSymbolType ANY f.open_symbol;
PopupItem <100> "Change Symbol Type..." Context (AllContexts ||
WantPopupMenus)
    TargetSymbolType ANY f.change_symbol;
PopupItem <100> "Symbol Properties..." Context (AllContexts ||
WantPopupMenus)
    TargetSymbolType ANY f.sym_desc;
PopupItem <100> "Delete" Context (AllContexts || WantPopupMenus)
    TargetSymbolType ANY f.delete_symbol;
PopupItem <100> "Hide" Context (AllContexts || WantPopupMenus)
    TargetSymbolType ANY f.hide_symbol;
PopupItem <100> "Set Star Center" Context (AllContexts || WantPopupMenus)
    TargetSymbolType ANY f.star_center_symbol;
PopupItem <100> "PopupSeparator1" Context (AllContexts || WantPopupMenus)
    TargetSymbolType ANY f.separator;
PopupItem <100> "Object Properties..." Context (AllContexts ||
WantPopupMenus)
    TargetSymbolType ANY f.obj_desc;

/*
 * Toolbar buttons with text
 */

#ifdef SPECIAL_EDITION
  /* Uncomment these for text buttons */

  /* ---- Remove this line for text buttons on the toolbar ----

  ToolbarButton <100> "Close" Context "AllContexts" f.close_smap;
  ToolbarButton <100> "Home" Context "AllContexts" f.goto_home;
```

```
ToolbarButton <100> "Root" Context "AllContexts" f.goto_root;
ToolbarButton <100> "Parent" Context "AllContexts" f.goto_parent;
ToolbarButton <100> "TBAR_SEP1" Context "AllContexts" f.separator;
ToolbarButton <100> "Pan" Context "AllContexts" f.show_panner;
ToolbarButton <100> "TBAR_SEP2" Context "AllContexts" f.separator;
ToolbarButton <100> "Open View" Context "AllContexts" f.about_ovw;

---- Remove this line for text buttons on the toolbar ---- */
#endif /* !SPECIAL_EDITION */
/*
 * Toolbar buttons with icons - icon paths are relative to OV      bitmaps
 * directory.
 */
ToolbarButton <100>
#ifdef NT
    @"toolbar/close.bmp,Close"
#else /* NT */
    @"toolbar/close.24.pm:toolbar/noclos.24.pm"
#endif /* NT */
    Context "AllContexts" f.close_smap;

ToolbarButton <100>
#ifdef NT
    @"toolbar/home.bmp,Home"
#else /* NT */
    @"toolbar/home.24.pm:toolbar/nohome.24.pm"
#endif /* NT */
    Context "AllContexts" f.goto_home;
ToolbarButton <100>
#ifdef NT
    @"toolbar/root.bmp,Root"
#else /* NT */
    @"toolbar/root.24.pm:toolbar/noroot.24.pm"
#endif /* NT */
    Context "AllContexts" f.goto_root;
ToolbarButton <100>
#ifdef NT
    @"toolbar/parent.bmp,Parent"
#else /* NT */
    @"toolbar/parent.24.pm:toolbar/noparent.24.pm"
#endif /* NT */
    Context "AllContexts" f.goto_parent;
ToolbarButton <90> "TBAR_SEP1" Context "AllContexts" f.separator;
ToolbarButton <90>
#ifdef NT
    @"toolbar/pan.bmp,Pan && Zoom"
```

## Controlling Map Access

### Using ARF Files to Control Menu Choices

```
#else /* NT */
    @"toolbar/pan.24.pm:toolbar/nopan.24.pm"
#endif /* NT */
    Context "AllContexts" f.show_panner;
    ToolbarButton <0> "TBAR_SEP2" Context "AllContexts" f.separator;
    ToolbarButton <0>
#ifdef NT
        @"toolbar/hpov.bmp,About HP OpenView"
#else /* NT */
        @"toolbar/hpov.24.pm:toolbar/nohpov.24.pm"
#endif /* NT */
    Context "AllContexts" f.about_oww;
}
```

## Allowing Others to View NNM from Many Computers

There are two ways to access NNM from multiple computers while it is running on your management station.

- Remote consoles  
Allows multiple people to use NNM as if they were sitting at the management station.
- NNM's web interface  
Allows multiple people to have read-only access to the maps and limited read-write access to the alarm tracking system. The web interface updates dynamically to display current network activity.
- Microsoft Terminal Services  
Allows multiple people to use NNM as if they were sitting at the management station.

### Remote Consoles

A number of people can view one map simultaneously or multiple maps by using remote console access (see *A Guide to Scalability and Distribution* manual for information). Remote consoles access the NNM services (background processes) and databases running on the management station, yet run NNM's user interface locally.

This is a great way to distribute the work load among your team. The map and alarms browser list are updated dynamically so everyone on your team knows the current network status. Everyone accessing NNM can acknowledge/delete alarms from the Alarm Browser as issues are resolved.

The number of remote access logins that can be supported is dependent upon a number of factors. See the *Performance and Configuration Guide* that was included with NNM's installation package.

Keep in mind that each remote station generates some network traffic in order to operate.

## NNM's Web Interface

Your team can remotely access the NNM session that is running on the management station through the web interface. See Chapter 14, “NNM on the Web,” for more information.

## Microsoft Terminal Services

In the Terminal Server multiuser environment, a terminal emulator allows a user to locally display the Windows 2000 desktop of a remote server and manage NNM on that remote server. To enable this functionality, the server must have Microsoft Terminal Services enabled and the local workstation must have Microsoft Terminal Services Client installed. See the *Installation Guide* for more information.

The Terminal Server machine becomes a core server that allows multiple users the ability to connect and display its desktop on their local machines via a Terminal Services Client application. A user can connect (via Terminal Services Client) to multiple Terminal Servers, or connect multiple times to the same Terminal Server and locally display multiple desktops.

The Microsoft Terminal Services client is a small executable that resides on each client workstation. Its function is to display the remote Windows 2000 user interface to the user.

See *A Guide to Scalability and Distribution* manual for more information.



## Closing All Current Sessions

Sometimes, in order to configure, update, or troubleshoot NNM, it is necessary to stop and start the NNM user interface (you *don't* need to stop all the services/background processes). To close all sessions (local, remote, and web) that your team is currently using, do the following:

1. To display a list of currently running sessions, at the command prompt, type:

```
ovstatus -v ovuispmd
```

You may want to print out this information so that you have the hostnames and session IDs for restarting the sessions later. You may wish to notify users who are currently accessing NNM to warn them about the shutdown.

2. To stop all sessions, at the command prompt, type:

```
ovstop -c ovuispmd
```

(This terminates any sessions and stops the `ovuispmd` service--the maps are all closed but your network is still being monitored by NNM services/background processes.)

3. To start new NNM sessions, at the command prompt, type:

```
ovstart -c ovuispmd
```

4. Now open NNM's user interface to begin one or more sessions.

Controlling Map Access  
**Closing All Current Sessions**

---

---

# 10

# Keeping Up with Events on Your Network: Beyond the Maps

Once you have verified the accuracy of NNM's maps of your network, you can be sure that communications are established with all important devices. It is now time to turn your attention to the second aspect of NNM, the event system.

In this chapter, you will find:

- *Background Information* (page 309).  
An explanation about how the NNM event system works.
- *NNM's Alarm Browser Overview* (page 315).  
The Alarm Browser provides a convenient central location for your team to monitor critical events on your network.
- *Alarm Browser Customization* (page 329).  
You can control the way the Alarm Browser looks and operates.

See Chapter 11, "Event Reduction Capabilities: Getting to the Root Cause," for information about the event reduction tools that are included with NNM. Also learn how you can use event correlation and the de-duplication feature to automatically determine the root cause of a problem, eliminate unuseful alarms, and in response to a pattern of events, generate new alarms that contain exactly the data you need for troubleshooting.

See Chapter 12, "Customizing Events: Doing It Your Way," for an explanation about how you can implement your own custom event monitoring strategies using NNM's Event Configuration, Application Builder, and Data Collection & Threshold features. You will also learn how to utilize vendor-specific MIBs and MIFs using NNM's MIB Load/Unload, MIB Browser, and DMI Browser features.

See Chapter 13, "Using Event Data," for an explanation about how you can take advantage of the information that has been gathered. NNM provides tools for using this information for analytical purposes and allows you to port the data to the application of your choice. You will learn about NNM's MIB Grapher and the data warehouse features.

## How NNM's Event System Works

Many services (background processes) within NNM and other HP OpenView compatible programs, gather information and generate **events** that are forwarded to NNM. Events can also be emitted from agents on managed nodes, or from management applications residing on the management station or on specific network nodes. Unsolicited SNMP events or notifications are called **traps**. NNM provides one centralized location, the Alarm Browser, where the events and traps are visible to your team. You can control which events and traps are considered important enough to show up as **alarms**. You and your team can easily monitor the posted alarms and take appropriate action to preserve the health of your network. Examples of alarms include:

- A critical router exceeded its threshold of traffic volume that you configured in NNM's Data Collection & Thresholds feature (`snmpCollect` service).
- The shell script that you wrote or the MIB application that you created using NNM's Application Builder feature gathered the specific information you needed and posted it to the Alarm Browser.
- An unauthorized network topology change was detected (`netmon` or `ovrepld` services).
- At 3 a.m., you receive a page notifying you that one of your mission-critical servers is using its UPS battery power. You received this page because of the way you configured the underlying DMI event `dmtfUPSBatteryUtilityPowerLostSystemOnBattery` (`ovcapsd` service and `ovactiond` service).
- The theft of a network printer was detected during NNM's regularly scheduled polling cycle (`netmon` service).
- An SNMP agent on a managed critical server forwarded a trap to NNM (`ovtrapd` service) because it was overheating and about to fail.

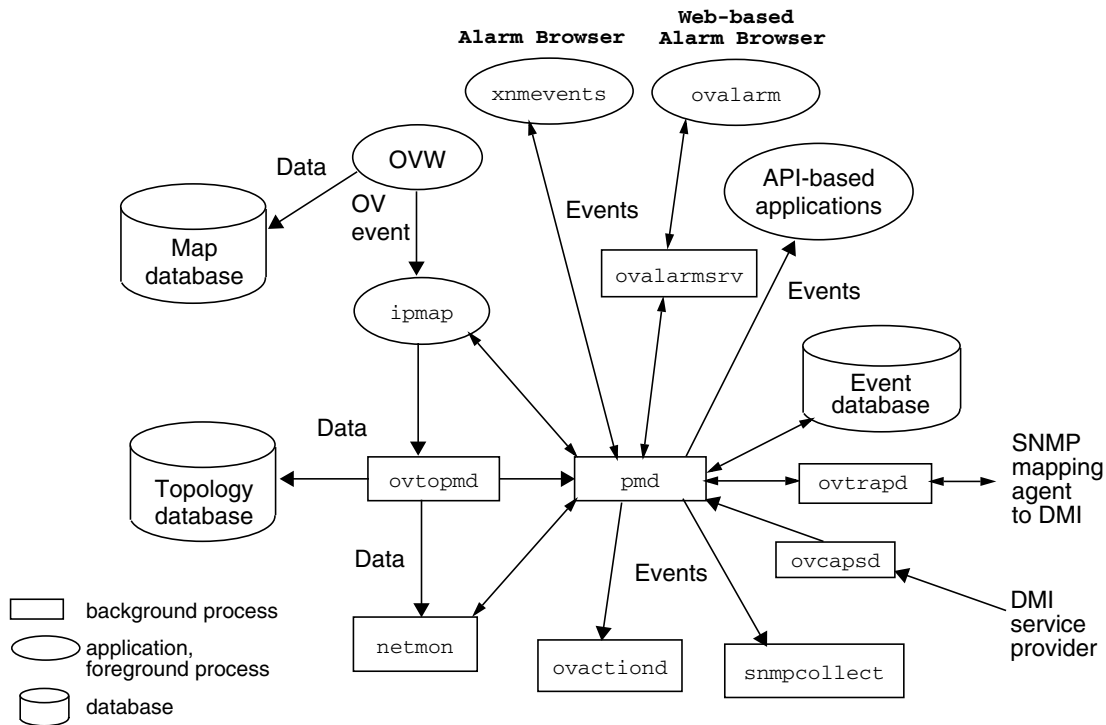
All events are passed to NNM's `pmc` service, which then logs them in the event database and sends them on to all applications that subscribe to them. For example, the Alarm Browser subscribes to all events configured to *display in a category* of the Alarm Browser.

**NOTE**

Alarms are not received by or displayed in the Alarm Browser if the `pmd` service is not running. See “Make Sure NNM’s Services Are Successfully Running” on page 89 for information about how to verify that `pmd` is running. Refer to Appendix A, “NNM Services and Files,” on page 523 for more information about these services (background processes).

If additional actions were configured to automatically execute upon the `pmd` services’ receipt of a certain alarm (such as dialing a pager or sending an email message), the alarm is also forwarded to NNM’s `ovactiond` service.

**Figure 10-1** `pmd` Interactions with Other NNM Services



By default, NNM’s Event Configuration feature includes preconfigured definitions for all standard traps. However, by default not all definitions are configured to post alarms into the Alarm Browser. Use NNM’s Event

Configuration feature to specify which traps are important to you and should, therefore, be posted in the Alarm Browser category of your choice (see “Event Configuration Overview” on page 404).

## SNMPv1 Traps / SNMPv2c Traps and Informs

An SNMP notification is a message sent from an agent to notify another system of an event on the local system. The notification may either be acknowledged (SNMPv2c *InformRequest*) or unacknowledged (SNMPv1 *TrapResponse* or *SNMPV2cTrap*).

NNM provides you with useful troubleshooting information each time a notification is received:

- The name or address of the node from which the notification came (also called an agent address)
- The notification identification (that is, trap number or notification object ID)
- Notification-specific variables (or data)

A **trap** is an unacknowledged notification sent from an agent to the management station without an explicit request from NNM. Agents can be configured to send traps to the NNM management station to indicate that a particular condition exists on the agent system, such as an error has occurred. Depending upon the SNMP agent, traps may be sent over and over until the problem is corrected on the attached node. An **inform** is an acknowledged notification sent from one management station to another management station. An inform requires a reply from the recipient. If no reply is received, the inform message is resent.

See the *snmpnotify* reference page in NNM's online help (or the UNIX manpage) for a description about how to generate these notifications, as well as how to identify the cause of a received notification.

## Ensuring that NNM Receives Traps from Your Network Devices

When configuring the SNMP agent on each network device, configure the agent's trap-forwarding list (or trap-destination list) to include the NNM management station's host name or IP address. Refer to the agent's documentation for information about how to do this. If the NNM management station is included on the trap-forwarding list, NNM receives notice from the agent when something goes wrong (even if the device does not show up on your NNM maps).

If your network device is using the HP SNMP EMANATE agent, NNM's netmon service automatically updates the agent's trap-forwarding list upon initial discovery, provided your network's SET-community names are configured within NNM. See "GET- and SET-Community Name and SNMP Port Issues" on page 128.

---

**NOTE**

For your convenience, NNM lets you view the currently configured trap-forwarding list on remote HP 9000 or Sun SPARCstations running the HP OpenView SNMP Agent software. On any submap, select the symbol for the HP 9000 or Sun SPARCstation running the HP OpenView SNMP Agent software, then do one of the following:

- Select the Configuration:SNMP Trap Recipients menu item.
- Use NNM's MIB Browser and the HP-UNIX MIB's object:  
    .1.3.6.1.4.1.11.2.13.1.2

(You must first configure your correct community names. See "GET- and SET-Community Name and SNMP Port Issues" on page 128.)

---

### **Sending SNMP Traps from the NNM Management Station**

To forward SNMP trap and inform messages originating from the NNM management station to another location of your choice, you must configure the SNMP agent on the NNM management station to include the destination host name or IP address in its trap-forwarding list. You must also manually configure the protocol versions that the destination supports. Use the `xnmsnmpconf -setVersions` command to configure this feature. Refer to the `xnmsnmpconf` reference page in NNM's online help (or the UNIX manpage) for more information.

For example, to configure protocol versions SNMPv1 and SNMPv2C for the default port for system `robert`, use the following command:

```
xnmsnmpconf -setVersions robert/162 1,2C
```

---

**NOTE**

For information about forwarding alarms from one management station to another management station, see *A Guide to Scalability and Distribution*.

---



## DMI Events (Indications)

The Desktop Management Interface (DMI) is a parallel management strategy to SNMP. DMI standards were developed by the Desktop Management Task Force (DMTF). Although DMI is entirely independent from the SNMP protocol, there are many similarities:

- The DMI **client** is similar to the SNMP management station.
- The DMI **service provider** is similar to the SNMP agent and must be running on each remote DMI device.
- The DMI **MIF** (Management Information Format) file is similar to the SNMP MIB file and defines the management information that can be provided by the service provider (event) or requested by the client (get/set).
- The **DMI event** is similar to the SNMP trap.

If your management station is running NNM on a Windows operating system, during discovery polling, NNM subscribes to receive DMI events from all managed systems running DMI version 2.0. All DMI events are received by NNM's `ovcapsd` service (background process) and converted to SNMP traps as specified in the Desktop Management Task Force's DMI to SNMP Mapping Standard. See the *ovcapsd* reference page in NNM's online help for more information.

---

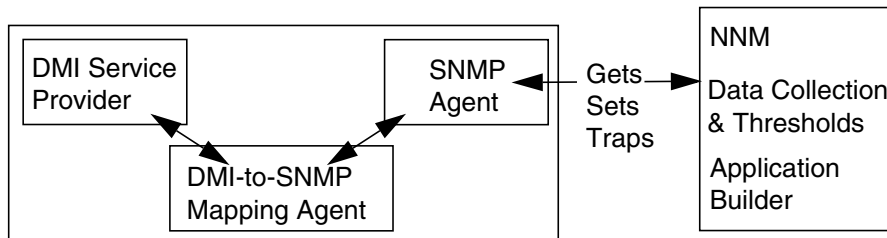
### NOTE

If your management station is running NNM for a UNIX system, you must communicate with the DMI service provider indirectly through an SNMP agent installed on the remote DMI-enabled device itself along with a DMI-to-SNMP mapping agent (such as Intel's). See the following figure.

The SNMP agent then handles all communications. Therefore, NNM doesn't need to subscribe to the DMI service provider in order to receive DMI events from this node.

---

**Figure 10-2** Remote DMI Enabled Node



All of the Desktop Management Task Force's standard DMI events that were available at the time of NNM's release are preconfigured in NNM's event system to post automatically to the Alarm Browser. You can customize the behavior of these events in the same manner that you can customize any SNMP event (see "Event Configuration Overview" on page 404 for information about customizing event configurations).

In addition to the DMTF's standard DMI MIF definitions, third-party vendors may have developed custom MIFs with their own custom events. NNM will receive these custom events and translate them to a generic DMI Event Received message until the vendor-specific DMI-to-SNMP mapping is configured in NNM. See "Loading DMI-to-SNMP Event Mappings" on page 401 for information about configuring a DMI-to-SNMP mapping for a custom MIF file.

## Alarm Browser Overview

NNM's Alarm Browser provides a convenient central location for monitoring critical events on your network. The Alarm Browser is useful for the following operations:

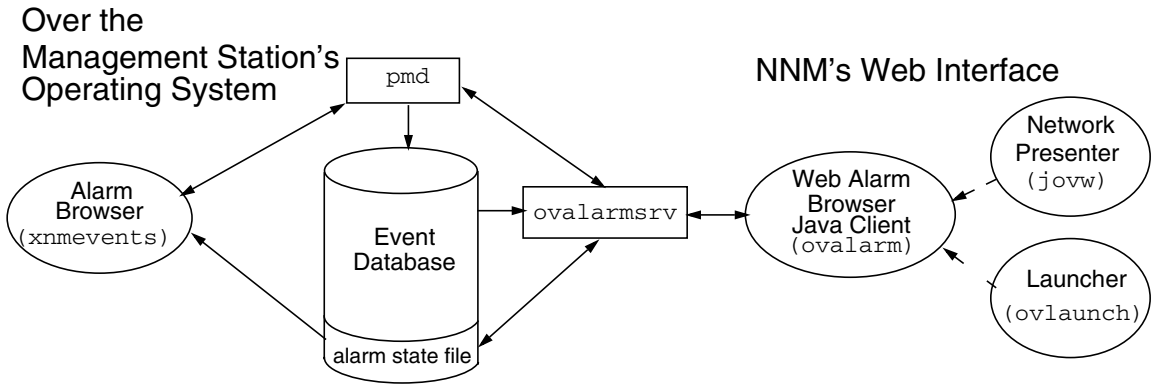
- Displaying useful information about an alarm.
- Sorting alarms into categories.
- Acknowledging that the problem causing an alarm is being addressed.
- Dynamically filtering the alarm list in multiple ways to make the information more useful; for example, by specific node.
- Deleting alarms from the list after they are resolved.
- Specifying additional actions that can be executed upon selected alarms.

There are actually two NNM Alarm Browser programs:

- one that runs under the management station's operating system
- one that runs within NNM's Java-based web interface

Both programs *share* the event database and configuration files for alarms as shown in the following figure. Therefore, no matter which Alarm Browser you use, the displayed list of alarms remains current at all times. Whenever someone acknowledges an alarm, deletes an alarm, or changes severity for an alarm everyone on the team sees the change.

**Figure 10-3 Relationship Between the Two Alarm Browsers**



## Displaying Alarms

To view alarms, open the NNM user interface, then:

- *Windows or UNIX:* Either,
  - In the Alarm Categories window, click on the button that corresponds to the category you wish to view.
  - In any submap, select the `Fault:Alarms` menu item to display the All Alarms Browser window. If any symbols are selected on your map, the alarm list is *filtered* to show only alarms related to the selected symbols.
  - In any submap, select the `Tools:Views->Home Base` menu item. Once NNM displays Home Base, select the Alarm Browser Tab. See “Overview of Dynamic Views and Home Base” on page 466 for more information.

- *Java-based web interface:* Either,
  - In the Launcher, select the Tools tab, then select NNM:Alarm Browser. In the Alarm Categories window, click on the button that corresponds to the category you wish to view.
  - In Network Presenter, select the Fault menu, then select Alarm Browser. In the Alarm Categories window, click on the button that corresponds to the category you wish to view.

To exit any of the Alarm Browser windows, select File:Close.

## Alarm Categories/Alarm Browser Windows

For your convenience, NNM sorts incoming alarms into categories. The categories may change depending on whether you purchased the NNM Starter Edition or the NNM Advanced Edition. The Alarm Categories window contains push buttons that correspond to each of the alarm categories. (The All category contains all the alarms that are present in the other categories.) Clicking on one of the push buttons displays the corresponding Alarm Browser window. You can add or change these categories (see page 335 for more information). Figure 10-4 shows the default Alarm Categories window from the NNM Advanced Edition.

Figure 10-4 Alarm Categories (Windows, UNIX, Web, and Home Base)




The push buttons in this window change color to indicate that alarms have been received. The color of the push button reflects the most severe unacknowledged alarm in the category, or it remains the background color if no alarms are present or all alarms are acknowledged. By default, alarm severity is indicated by the following colors:

Normal	Green
Warning	Cyan
Minor	Yellow
Major	Orange

Critical	Red
No alarms	Background color
All alarms acknowledged	White

When you select a category, the corresponding Alarm Browser window is displayed. The Alarm Browser window contains a scrollable list of alarms belonging to the associated category. The Alarm Browser window also displays status about the number of alarms in the window and their severity. Figure 10-6 shows the All Alarms Browser window.

To view details about alarms:

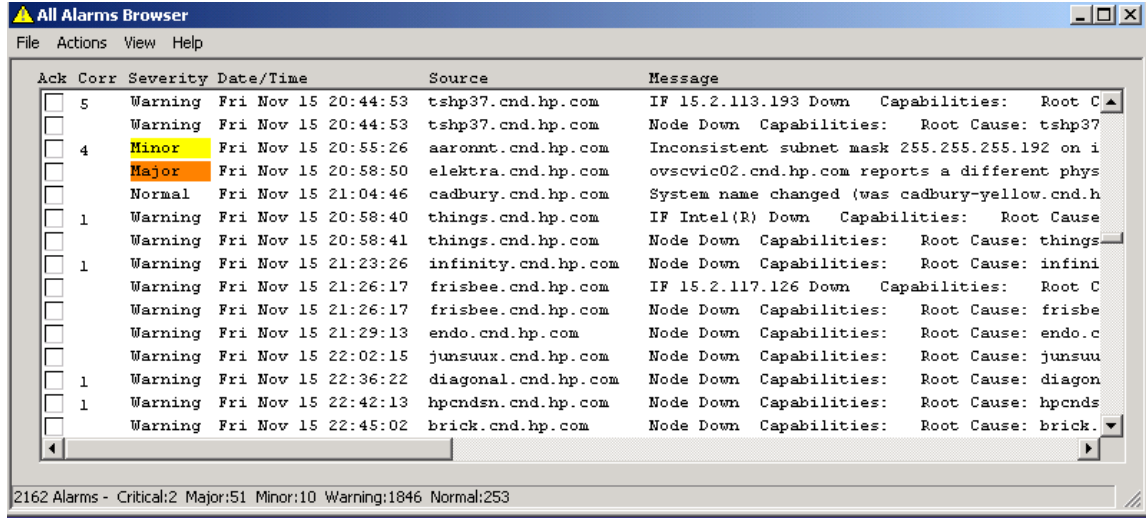
- *Windows or UNIX:*
  - Click the right mouse button on any alarm message to display a pop-up window that contains a summary.
  - Select the `Actions:Alarm Details` for complete details.
- *Java-based web interface:*
  - Click on the  button for complete details.
  - Select the `Actions:Alarm Details` for complete details.

It is possible to sort the alarm list:

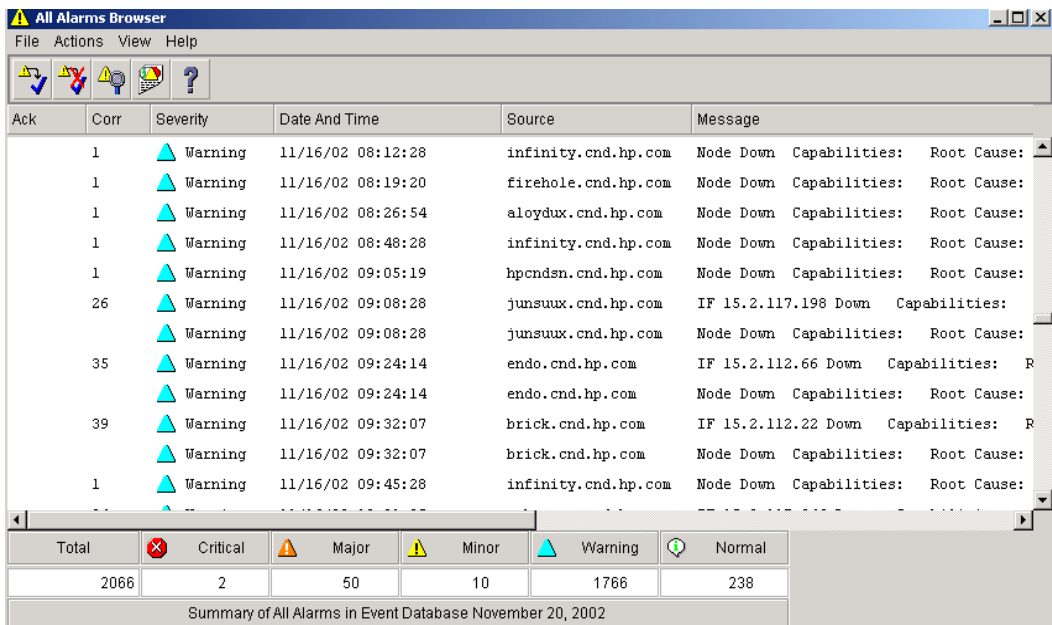
- *Windows or UNIX:* Select `Actions:Additional Actions` for sorting choices.

- *Java-based web interface:* Click once or twice on any column heading.

**Figure 10-5 All Alarms Browser Window (Windows or UNIX)**



**Figure 10-6 All Alarms Browser Window (Web)**





The Alarm Browser window lists all undeleted alarms for the indicated category. By default, the alarms are in chronological order with the most recent event at the *bottom* of the list.


For each alarm, the list displays the following:

Ack	A check mark indicating whether the alarm is acknowledged or unacknowledged.
Corr	A number in this column indicates that this is a primary (root cause) alarm for a group of alarms identified by the Event Correlation System. This number represents the quantity of related or duplicate events correlated beneath the most recent alarm.  If you double-click on this number, NNM displays all of the events nested under the alarm.
Severity	The severity of the alarm.
Date/Time	The day of the week, date, and time when the alarm was received.
Source	An identifier (such as a node name) for the network object where the alarm originated.
Message	A brief description of the alarm.

## Acknowledging Alarms

In order to keep track of which alarms are being worked on, members of your team can acknowledge an alarm, yet not delete it. When an alarm is acknowledged, everyone who has access to the Alarm Browser sees the change. This is a great way for your team to communicate about which issues are being addressed.

Once acknowledged, an alarm remains visible but does not affect the color propagated into the Alarm Categories window. To acknowledge selected alarms:

- *Windows and UNIX:*  
Either select the `Actions:Acknowledge` menu item or click in the Ack field.
- *Java-based web interface:*  
Either select the `Actions:Acknowledge` menu item or click the  button.

## Filtering Alarms

Filtering allows you to limit the number of alarms displayed. You display only the alarms that are meaningful to your particular troubleshooting efforts at the time; for example, all alarms related to a specific node. Your filtering choices do not affect the Alarm Browser lists being viewed by other members of your team. When you close any Alarm Browser window, the filters are automatically cleared.

Each person using NNM can dynamically filter their displayed alarm list based upon one or more criteria. The criteria must all be true for an alarm to display in the list:

- Severity level
- Source IP addresses or use wildcards to specify a range of IP addresses or node names
- Acknowledged or unacknowledged alarms
- Alarm time span
- Message string word search
- Event type

Filters are available from any Alarm Browser window under `View:Set Filters`.

- *Windows or UNIX:*  
If filtering is on, the status message at the bottom of the Alarm Browser window indicates that filtering is applied.

Each Alarm Browser window can use unique alarm filter settings. Only one filter dialog box can be displayed at a time. The title bar of the filter dialog box indicates which Alarm Browser window is currently affected. If the filter dialog box is displayed and the `View:Set Filters` menu item is selected from a different Alarm Browser window, the name in the filter dialog box title bar changes and the filter settings in the dialog box change to reflect current filter settings for that Alarm Browser window.

You can perform operations based upon the filtered alarms, the selected alarms, or all the alarms in the category. For example, to see all the Critical alarms, pick the All Alarms category. From the All Alarms Browser window pick `View:Set Filter`, then ensure that only Critical severity is selected and click [Apply] or [OK]. The filter modifies the list of alarms currently displayed in the Alarm

Browser window. You now see all the critical alarms. After resolving all these alarms, delete them using `Actions:Delete->Filtered Alarms`. The category light in the Alarm Category window changes from red to the next most severe color.

You can save and restore the filters you have set by using the [Save] and [Restore] buttons in the Set Filters dialog box (not available in the web-based Alarm Browser).

- *Java-based web interface:*

If filtering is on, the status message at the bottom of the Alarm Browser window indicates that filtering is applied.

You can perform operations based upon the filtered alarms and the selected alarms. For example, to see all the Critical alarms, pick the All Alarms category. From the All Alarms Browser window pick `View:Set Filter`, then ensure that only Critical severity is selected and click [Apply] or [OK]. The filter modifies the list of alarms currently displayed in the Alarm Browser window. You now see all the critical alarms. After resolving all these alarms, delete them using `Actions:Delete->All in Browser`. The category light in the Alarm Category window changes from red to the next most severe color.

Each Alarm Browser window can use unique alarm filter settings. Select the `View:Set Filters` menu item from any Alarm Browser window to set its filter.

---

**NOTE**

Two additional alarm filters are available on a user-by-user basis; they need to be implemented on each management station and each remote console. These filters are not available through NNM's web interface:

- `filterByMap`  
Display only alarms that apply to devices on the current map.
- `filterByMapManaged`  
Display only alarms that apply to managed devices on the current map.

These two filters are invoked through settings in:

- *Windows:* The system registry settings for `xnmevent`  
See the *app-defaults* reference page in NNM's online help.

- *UNIX*: `$APP_DEFS/XNmevents` file  
See the comments within this ASCII file.
- 

## Deleting Alarms

Alarms remain visible in an Alarm Browser window until someone deletes them or until the specified maximum number of alarms for the browser has been reached. If you do not delete (or acknowledge) alarms, each button in the Alarm Categories window will continue to show status color for all alarms, and you may not know when new alarms occur. When an alarm is deleted, it is removed from everyone's alarm list.

You can delete alarms either individually or all at once, using choices under the `Actions:Delete` menu item in each Alarm Browser window. Whenever you delete alarms, a confirmation window pops up.

Once deleted, the same alarm instance does not reappear. You need only delete the alarm in one Alarm Browser window. This clears the alarm from every Alarm Browser window in which it appears.

See “Controlling the Size of the Alarm Browser’s State File” on page 331 for information about controlling the maximum number of alarms; the default is 3500.

---

### NOTE

If you want to turn off the confirmation dialog box:

- *Windows*: In the system registry, set `XNevents/warnOnDelete` to false. See the *app-defaults* reference page in NNM’s online help.
- *UNIX*: In the `$APP_DEFS/XNevents` file, set `warnOnDelete` to false. See the comments in this ASCII file for more information.

The confirmation dialog box does not appear over the web-based interface.

---

---

### TIP

If you plan to delete a large number of alarms and wish to be able to undo the delete, see “Copy or Restore the Alarm Browser’s State File” on page 334.

---

## NNM's Map and the Alarm Browser

The map and the Alarm Browser are tightly integrated. When you double-click on an alarm from the Alarm Browser, NNM displays the source node in the submap, zooms to the highlighted object, and displays the panner window. Conversely, from a submap, NNM makes it easy to display all alarm messages related to selected symbols.

### Highlighting an Alarm Message and Displaying the Relevant Submap

---

**NOTE**

Selecting alarm messages in an Alarm Browser window does not affect the current object selections on your map.

---

To jump directly from an alarm message to the corresponding submap, select the alarm in the Alarm Browser window. Select:

- *Windows or UNIX:* Actions:Highlight Source on Map or simply double-click on the alarm message. The submap containing the source of the alarm is displayed. The label for the source object is *highlighted*. Any map symbols that were previously highlighted or previously *selected* remain so.
- *Java-based web interface:* Actions:Show Node in Network Presenter. If accessing a map other than the *default* map, you must type in the correct map name before proceeding. A new instance of the Network presenter is started. Any previously running instance of the Network Presenter is closed. The submap containing the source of the alarm is displayed. Only the symbol for the source object is *selected*.

On the submap, use NNM's menu items to obtain more information to help you resolve the problem that has caused the alarm.

---

**TIP**

You may receive alarms for nodes that do not appear on the map. For example:

- If an SNMP agent is not in your management region, but is configured to send traps to the NNM management station, the Alarm Browser receives alarms for that agent (for example, when it starts up).

- If you are displaying a map's snapshot, the alarms are for the current map itself, not the snapshot.
- 

### Selecting Map Symbols and Displaying Associated Alarm Messages

To access alarm information about objects on the map:

- *Windows or UNIX:* Select the objects on the map and then select `Fault:Alarms`. The `All Alarms Browser` window displays automatically filtering out all alarms except those associated with the selected objects.
- *Java-based web interface:* From any submap, select the `Fault:Alarms` menu item; or select an object on the map, then right-click and select `Alarm Browser`. The `Alarm Category` window is displayed. To view alarms for specific devices, open the `All Alarms` list and use `View:Set Filters` to limit the alarm list.

### Specifying Additional Actions on Alarms

You can select alarms and pipe the alarms into user-written programs (shell commands via standard input), using the menu item `Actions:Additional Actions`. The output of the command will be displayed.

---

#### NOTE

This feature is not available in the web-based Alarm Browser.

---

Only one `Additional Actions` dialog box can be displayed at a time. The `Additional Actions` dialog box affects alarms in the browser that is indicated in its title bar. If this dialog box is displayed and the `Actions:Additional Actions` menu item is selected in a different `Alarm Browser` window, the name in the title bar changes to reflect the alarm category for the most recently selected browser.

Specify the scope of the action. You can select one of five different scopes for the action. These are:

- Selected Alarms
- Filtered Alarms

- Acknowledged Alarms
- Unacknowledged Alarms
- All Alarms in Category

Clicking [Describe] when an action is highlighted provides a description of the action, including what command is executed, and how long NNM waits for the command to complete before terminating the command.

See the *trapd.conf* reference page in NNM's online help (or the UNIX manpage) for an explanation about how to write additional action commands. The command is a program that expects input in a predefined format. See "Defining Additional Actions" on page 423, which describes how to define additional actions.

## Launching Specific Views or URLs from Alarms

You can configure NNM to access a specific web location from an event in the Alarm Browser. You can configure up to 50 URLs in the *xnmeventsExt.conf* file for the `Actions:Views` menu. You can already access Dynamic Views from Alarm Browser events, however these new URLs can be any other URL that you prefer. NNM views are available from NNM's `Tools` menu.

To configure an alarm to launch a specific URL, edit the following file:

- **Windows:** *install\_dir*\conf\C\xnmeventsExt.conf
- **UNIX:** `$OV_CONF/$LANG/xnmeventsExt.conf`

Editing instructions are included in the file.

After you modify *xnmeventsExt.conf*, do the following for the changes to take effect:

1. Close the NNM Alarm Browser.
2. Stop the `ovalarmsrv` process:
  - **Windows:** Run the *install\_dir*\bin\ovstop `ovalarmsrv` command.
  - **UNIX:** As root, run the `$OV_BIN/ovstop ovalarmsrv` command.

3. Start the `ovalarmsrv` process:
  - **Windows:** Run the `install_dir\bin\ovstart ovalarmsrv` command.
  - **UNIX:** As root, run the `$OV_BIN/ovstart ovalarmsrv` command.
4. Run the `xnmevents` command to restart the NNM Alarm Browser:
  - *Windows:* Run the `install_dir\bin\xnmevents.exe` command.
  - *UNIX:* Run the `$OV_BIN/xnmevents.exe` command.
5. Restart any web-based Alarm Browsers to view the new menu items.
6. Use the `Fault:Alarms` menu item to open the Alarm Browser. The `Actions:Views` menu in the Alarm Browser should contain the new menu items.

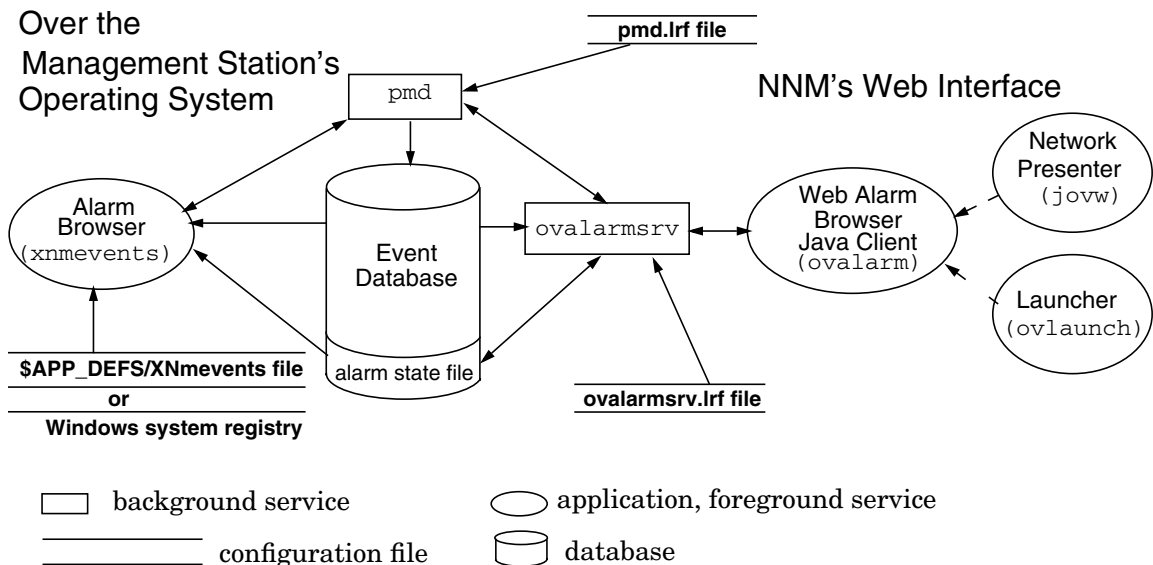


## Configuring the Alarm Browser

You can configure the Alarm Browser in the following ways:

- Set the maximum size of the event database (default is 16MB).
- Set the maximum number of alarms in the Alarm Browser's state file (default is 3500).
- Set the number of alarms that should be deleted each time the state file reaches maximum capacity.
- Copy or restore the Alarm Browser's state file.
- Control the look and behavior of the Alarm Browser.
- Assign alarm categories:
  - Moving alarms from one category to another.
  - Changing the default alarm category for a specific MIB object.
  - Creating new alarm categories and assigning alarms to the new category.

**Figure 10-7 Configuration Points in the Two Alarm Browsers**



## Controlling the Size of the Event Database

All events are entered into the event database. The most important ones are posted to the Alarm Browser. Many NNM services rely upon the information stored in the event database. The default maximum size setting is 16MB. The event database is divided into four files. In the default configuration of 16MB, this means that each file has a maximum size of 4MB. When all four files are full, the oldest log is truncated and new events are written into the reclaimed space.

You can change the amount of disk space that is reserved on the management station for the event database by setting the `b` parameter in the `pmd.lrf` file. See the `ov_event` and `pmd` reference pages in NNM's online help (or the UNIX manpages) for information about changing this setting. Information from the event database can be archived into the data warehouse for trend analysis by selecting the menu item `Tools:Data Warehouse->Export Events`.

---

### NOTE

The event database is accessed by all versions of NNM running on your management station: NNM for Windows *or* NNM for UNIX, and NNM on the web.

---

---

### TIP

Previous releases of NNM created log files called `trapd.log` and `trapd.log.old` rather than the event database. If your team developed special processes that depend upon the `trapd.log` file, it is possible to continue using it:

- The `trapd.log` file will be generated *in addition to*, rather than *instead of* the event database. You must change the `pmd.lrf` file to include the `-SOV_EVENT;t` command. See the `ov_event` reference page in NNM's online help (or the UNIX manpage) for more information. Generating the `trapd.log` file in addition to the event database requires space on your management station's hard drive.
  - If you wish to generate a `trapd.log` file occasionally, from the command line, type `ovdumpevents`. See the `ovdumpevents` reference page in NNM's online help (or the UNIX manpage) for more information.
-

## Controlling the Size of the Alarm Browser's State File

Events that are configured to post as alarms in the Alarm Browser are sent to the Alarm Browser's state file. The contents of this state file provides the starting point for displayed alarms each time you open the NNM interface. The state file maintains all user edits to the alarm list, such as acknowledgments or deletes.

To modify the number of alarms being stored and displayed, you must change the setting in two places.

- The `-a` parameter in the `ovalarmsrv.lrf` file.  
Changes in the `ovalarmsrv.lrf` file control what is stored in the state file. Remember that this setting affects all users who have access to NNM. Consider RAM and hard drive space on your management station when changing this setting.
- The `maxEvents` setting.  
This parameter is in the UNIX's `$APP_DEFS/XNmevents` file (English locale) or in the Windows system registry's `xnmevents` entry. It controls how many messages display in the Alarm Browser window itself (independent of the number stored in the underlying state file).

---

### NOTE

For other locales on UNIX operating systems, look for the `XNmevents` file in `/usr/lib/X11/locale/XNmevents`.

---

To change the `ovalarmsrv.lrf` file setting that controls the state file:

1. Make your changes to the `-a` parameter and save your changes:

- *Windows:* `install_dir\lrf\ovalarmsrv.lrf`
- *UNIX:* `$OV_LRF/ovalarmsrv.lrf`

See the *lrf* reference page in NNM's online help (or the UNIX manpage) for more information.

2. Update NNM's configuration file. At the command prompt, type:  
**ovaddobj ovalarmsrv.lrf**

3. To force NNM to acknowledge the change, at the command prompt, type:  
**ovstop ovalarmsrv**

---

**TIP**

---

This causes the Alarm Browser to close in all currently running NNM sessions.

Then at the command prompt, type:  
**ovstart.**

To change the `maxevents` setting that controls the display of messages, on each management station and each remote console (not required for the web-based Alarm Browser):

1. Make your changes to the `maxEvents` parameter and save your changes:
  - *Windows:* The system registry  
`MKEY_LOCAL_MACHINE:SOFTWARE:Hewlett-Packard:OpenView:Network Node Manager:xnmevents:maxEvents`
  - *UNIX:* `$APP_DEFS/XNmevents` file
2. To force NNM to acknowledge your changes, simply close the Alarm Browser; then reopen the Alarm Browser.

See the *ovalarmsrv*, *app-defaults*, and *xnmevents* reference pages in NNM's online help (or the UNIX *ovalarmsrv* and *xnmevents* manpages) for more information.

---

**TIP**

---

Previous releases of NNM used per-user state files (`xnmevents.user`). Do not try to restore these old state files into this version of NNM.

## Controlling How Many Alarms to Delete Automatically

Each time the Alarm Browser's state file reaches maximum capacity, the oldest alarms (at the top of the list) are deleted to make room for the new ones.

To control the number of alarms deleted each time, you must change the setting in two places.

- The `-d` parameter in the `ovalarmsrv.lrf` file.  
Remember that this setting affects all users who have access to NNM. Consider the frequency of pause time required to rewrite the state file's alarm list if you change this setting. The higher the number the less frequent the pause.
- The `deleteNumber` setting.  
This parameter is in UNIX's `$APP_DEFS/XNmevents` file or in the Windows system registry's `xnmevents` entry. It controls how many messages are deleted each time the Alarm Browser window itself reaches maximum size (independent of the number stored in the underlying state file).

To change the `ovalarmsrv.lrf` file setting that controls the state file:

1. Make your changes to the `-d` parameter and save your changes:

- *Windows:* `install_dir\lrf\ovalarmsrv.lrf`
- *UNIX:* `$OV_LRF/ovalarmsrv.lrf`

See the *lrf* reference page in NNM's online help (or the UNIX manpage) for more information.

2. Update NNM's configuration file. At the command prompt, type:

```
ovaddobj ovalarmsrv.lrf
```

3. To force NNM to acknowledge the change, at the command prompt, type:

```
ovstop ovalarmsrv
```

---

**TIP**

This causes the Alarm Browser to close in all currently running NNM sessions.

Then at the command prompt, type:

```
ovstart
```

To change the `deleteNumber` setting that controls how many messages are deleted from the Alarm Browser display window, on each management station and each remote console (not required for the web-based Alarm Browser):

1. Make your changes to the `deleteNumber` parameter and save your changes:
  - **Windows:** In the system registry, add the following parameter and specify the number:  
`MKEY_LOCAL_MACHINE:SOFTWARE:Hewlett-Packard:OpenView:Network Node Manager: xnmevents:deleteNumber`
  - **UNIX:** `$APP_DEFS/XNmevents` file
2. To force NNM to acknowledge your changes, simply close the Alarm Browser; then reopen the Alarm Browser.

See the *ovalarmsrv*, *app-defaults*, and *xnmevents* reference pages in NNM's online help (or the UNIX *ovalarmsrv* and *xnmevents* manpages) for more information.

## Copy or Restore the Alarm Browser's State File

The state file records all actions that have been applied to specific alarms, such as acknowledging an alarm, deleting an alarm, changing an alarm's category, or changing an alarm's severity rating. Remember that the state file affects all users who access NNM (unlike the filtering or sorting of alarm messages, which is controlled by each individual user).

There may be times when you want to delete a large number of alarms, yet be able to change your mind and undo the deletion. This is possible if you first save the state file.

NNM provides a tool for you to use that makes sure all services (background processes) are in the correct state, then copies or restores the state file for you.

See the *ovalarmadm* reference page in NNM's online help (or the UNIX manpage) for information about copying and restoring the Alarm Browser's state file.

---

### CAUTION

Do not work directly with the encoded binary state file:

- **Windows:** `install_dir\databases\eventdb\statelog`
- **UNIX:** `$OV_DB/eventdb/statelog`

All changes to this file are made through the Alarm Browser.

---

## Controlling How the Alarm Browser Looks

If there are compelling reasons that you need to change aspects of the Alarm Browser such as the color used to indicate the severity of alarms, explore the possibilities in:

- *Windows*: The system registry  
See the *app-defaults* reference page in NNM's online help for more information.
- *UNIX*: `$APP_DEFS/XNmevents` file  
See the information included in this ASCII file for more information.

## Assigning Alarm Categories

You can control the category into which an alarm message is displayed by:

- Moving an existing alarm message to another category (affecting only that instance of the particular alarm).
- Changing the default category for alarms received from a specific MIB object.
- Creating an entirely new alarm category and assigning alarms to the new category.

### Moving an Existing Alarm Message to Another Category

To move an alarm to a different category, in any Alarm Browser window, select the alarms that you want to recategorize. Then select `Actions:Assign Category` from the menu bar. Refer to the online help if you need more information.

This change only affects the selected instances of the alarms. Future alarms of the same type will continue to be posted in the old category's list, unless you reconfigure the alarm as described next.

### Changing the Default Category for an Alarm

---

**NOTE**

You cannot do this through NNM's web interface.

---

To permanently change the category assignment for a specific alarm message, you must change the configuration settings for the specific underlying event's MIB object. Access the `Event Configuration` dialog box:

- From any Alarm Browser window, select the alarm you wish to reconfigure, and select the `Actions:Configure Event` menu item. The `Event Configuration` dialog box displays the underlying event's current configuration.
- From any submap, select the `Options:Event Configuration` menu item. It is up to you to navigate to the MIB object that defines the event you wish to reconfigure.

Prerequisites are:

- The MIB for which you want to configure events must be loaded into NNM's MIB database. The `Options:Load/Unload MIBs:SNMP` operation lists all MIB modules currently in NNM's MIB database. If the enterprise-specific MIB is not loaded, see "Loading MIBs in the MIB Database" on page 395 for more information.
- Make sure you understand the enterprise-specific MIB. To configure events, you need to understand the trap definitions for the device and what they do. Most device vendors include documentation about their enterprise-specific traps with their product. This documentation typically describes the trap and when the trap is generated. In addition, some vendors include overview information describing strategies for how to manage their specific device. The vendor documentation can give you the conceptual understanding that you need to configure effective event formats and actions.

Refer to "Event Configuration Overview" on page 404 for information about reconfiguring an event to always post its alarm into a specific category of the Alarm Browser.

### Creating a New Alarm Category

---

**NOTE**

You cannot create a new alarm category through NNM's web interface. However, once you establish a new alarm category, it will show up in the web-based Alarm Browser.

---



You can configure NNM's Alarm Browser to contain custom categories that fit your organization's needs.

For example, you could add a category called `Link Address Alarms`. Once added, you could change all SNMP link address events so that new alarms of this type are posted into the new category.

To add a new category:

1. Access the Event Configuration dialog box from either any Alarm Browser window (Actions:Configure Event) or from any submap (Options:Event Configuration).
2. From the Event Configuration menu bar, click Edit:Alarm Categories. The Alarm Categories window appears.
3. Check NNM's online help that is available within the Alarm Categories window for more information.
4. In the Event Configuration window, click File:Save to save the changes you have made. The new event category appears in the Alarm Categories window.

---

**NOTE**

You must now assign specific SNMP events (associated with specific MIB objects) to this new category; otherwise, it will remain empty. For information about assigning specific events to this category, see "Event Configuration Overview" on page 404.

---



---

**11**

**Event Reduction Capabilities:  
Getting to the Root Cause**

## NNM's Event Reduction Capabilities

Event reduction is the process by which NNM can identify relationships between events. Once identified, a smaller number of new events with the same or higher information content can be generated. This simplifies diagnosing network faults. There are three event reduction strategies within NNM:

- “De-Duplication of Alarms” on page 347

Multiple, identical alarms are often sent to the NNM Alarm Browser. NNM can retain these alarms, but not display all of them by using *de-duplication* to nest the duplicate alarms beneath the most recent alarm. By reducing the quantity of alarms displayed in the Alarm Browser, you can easily determine the important alarms. You control which events are nested in this manner by making entries in the `dedup.conf` file.

- “ECS Correlations” on page 349

NNM includes built-in event correlation logic that evaluates incoming events and nests certain related events under a single meaningful alarm. Event correlation helps you quickly identify the primary cause of the problem. For example, you don't need to sift through hundreds of alarms to identify the root cause when a router goes down. High-speed correlation minimizes network bandwidth utilization and improves the quality of customer network services. The fundamental design principle is to correlate events in real-time.

- “Correlation Composer Correlators” on page 371

NNM provides additional built-in event correlation logic, called correlators, through the Correlation Composer. The Correlation Composer provides an easy-to-use interface for configuring and creating correlation rules.

The Correlation Composer can be launched in one of two modes:

- default (operator) mode for viewing and limited editing of correlators, and
- developer mode for creating and modifying correlators.

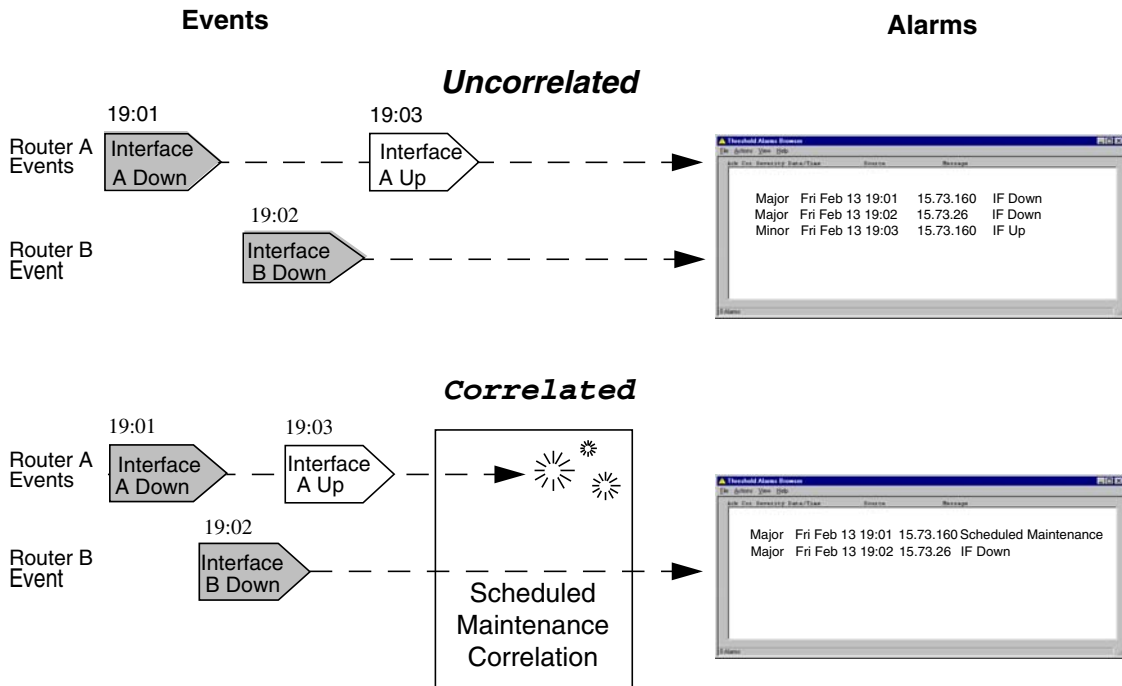
For more information about the two modes, access the following pdf format manual from the NNM main window. Click [Help:Documentation](#) and select *HP OpenView Correlation Composer's Guide*.

All three of the event reduction strategies determine patterns of events by monitoring the *SNMP MIB Object ID*, the sending *SNMP-agent/IP-address*, and a specific combination of *SNMP variable-binding values*. See “What is an SNMP Variable-Binding and How Do I Identify One?” on page 387 to learn about determining SNMP variable-binding values.

**Event reduction** modifies the flow of events by recognizing patterns of events, then either nesting related events in the Alarm Browser, discarding the events, or replacing the events with fewer more meaningful events. Event reduction strategies can dramatically reduce the number of alarms and improve the value of alarms displayed in your Alarm Browser. Instead of displaying the whole event storm typically generated by equipment and link failures, a correlated event stream displays only the most meaningful alarms, resulting in faster and easier identification of network problems.

For example, one of the correlations included with NNM is called the ScheduledMaintenance correlation. Once configured, it suppresses router events generated during routine maintenance.

**Figure 11-1 With/Without Scheduled Maintenance Correlation**



In Figure 11-1 on page 342, Router A is undergoing planned maintenance (for example, it is being moved to a new rack). The ScheduledMaintenance correlation has been configured to post a Scheduled Maintenance alarm notification and suppress all other events generated by this device during the maintenance period. Because the device has powered down within the maintenance period, the correlation suppresses both events from Router A. However, the Interface Down event from Router (Interface) B is passed through because it is not one of the devices specified for maintenance. The result is two events instead of three, and a higher confidence level that the alarm represents a real problem requiring attention from your team.

This simple correlation eliminates false alarms caused by maintenance activities. Where previously an automated paging facility would have resulted in unnecessary pages, with event correlation it may be feasible to generate meaningful pages.

As illustrated by this example, a single correlation recognizes event patterns caused by a particular fault type. Once you understand how a correlation works, you can modify its parameter values.

A detailed description of each of the supplied correlations and correlators is provided. Before changing any parameter settings, carefully read the description in the configuration window to learn how each correlation and correlator works, its effect, and its deployment requirements.

## Correlation Concepts

This section discusses the following concepts:

- “Enabling/Disabling Event Correlations” on page 343.
- “Static and Dynamic Parameters” on page 344.
- “Multiple Correlations and Correlators” on page 344.
- “Event Streams” on page 345.
- “Recommendations” on page 345
- “Event Reduction in Distributed Environments” on page 346.

### Enabling/Disabling Event Correlations

When you install NNM, several correlations and correlators are loaded and **enabled**, meaning that they are actively filtering incoming events.

Correlations and correlators can be enabled and disabled at any time through the ECS Configuration window and Correlation Composer window.

Many correlations work by comparing incoming events with the previously gathered history of those events. When a correlation is first enabled, it has no history and takes a certain amount of time to build one. In the meantime, events that would otherwise be suppressed may be passed through, and events that would normally be modified or generated may not be. When a correlation is first enabled, a certain amount of time is required before the internal tables fill with events so that the correlation can work consistently. This is called **settling time**.

---

**NOTE**

Some correlations do not provide any benefit until they are configured. For example, the `ScheduledMaintenance` correlation doesn't do anything until you specify the devices involved and a time span.

---

### Static and Dynamic Parameters

Parameters are classified as either static or dynamic. The distinction is not important when you are changing the parameters of a disabled ECS correlation. However, if the ECS correlation is already enabled, then the distinction affects the way in which the changes are applied.

When changes to **static** parameters are applied, the correlation is disrupted because the correlation must be disabled and re-enabled, and you must allow for settling time before the correlation is effective. On the other hand, changes to **dynamic** parameters take effect as soon as they are applied.

---

**NOTE**

The parameters of correlators within the Correlation Composer are all *dynamic*. Changes take effect as soon as they are applied.

---

### Multiple Correlations and Correlators

When multiple correlations and correlators are enabled in the *default* stream (see next section), an incoming event will be posted into the Alarm Browser only if *all correlations and correlators* that are configured to receive that event, pass that particular event through. In other words, if *any* correlation or correlator filters out the incoming event, then that event will not appear in the Alarm Browser, even if another correlation or correlator would otherwise have posted it as an alarm.

This point is especially important when you introduce a new correlation or correlator. You must make sure that the new one does not break any existing event reduction configurations: de-duplication, ECS correlations, and Correlation Composer correlators.



## Event Streams

A **stream** is a logically separate flow of events. All streams originate from a common source (the postmaster, `pmd` service). However, each stream can have a different set of correlations applied to it and is directed to a different destination.

NNM listens only to the *default* stream. You may wish to establish separate streams for special network management applications that you add to your OpenView suite. The ECS correlations can be used on the new streams. However, the Correlation Composer correlators only work on the default stream; do not enable them on any other stream.

If you configure NNM's ECS feature with multiple streams, you can choose different combinations of your correlations for each stream. Each correlation can be enabled only once on a selected event stream. However, you can enable the same correlation on several different event streams. *The configuration settings for the correlation remain the same across all streams.* Refer to the *ecsmgr* reference page in NNM's online help (or the UNIX manpage) for more information.

## Recommendations

The most detailed source of information about each correlation or correlator is the descriptive information displayed in each correlation's configuration window. Review this information before using the correlation or correlator.

To minimize disruption to the event reduction service, it is suggested that you:

- Plan changes ahead of time where possible.
- Choose a time when network activity is low.
- Do not make ad hoc changes to static parameters.
- Avoid inconsistent states by applying all parameter modifications at the same time, rather than applying each individual parameter change.
- Avoid changing static parameters in the middle of an event storm.
- Test the effect of the event reduction strategy to be sure it does what you expect.
- Ensure that all members of your team know which event reduction strategies are in effect.

### **Event Reduction in Distributed Environments**

The NNM correlation engine feeds information for all event reduction features into the ECS default event stream. Using NNM Advanced Edition in a distributed environment this has two main benefits:

- The correlation processing load is distributed over several collection stations.
- Network traffic is reduced.

See *A Guide to Scalability and Distribution* manual for further information about distributed network management concepts.

---

## De-Duplication of Alarms

NNM de-duplication affects only those alarms listed in the `dedup.conf` file. The *newest* de-duplicated alarm always becomes the parent alarm, visible in the Alarm Browser. Only those alarms that are displayed in the Alarm Browser are monitored by de-duplication. In other words, de-duplication runs on alarms after they have passed through the event correlation system.

---

### NOTE

All de-duplication configurations, correlators defined within Correlation Composer, and all correlations defined within ECS must *work well together* on your NNM management station. Before you begin, it is critical that you print out and read the following white paper:

Developing\_NNM\_Event\_Reduction.pdf

- *Windows:* `NNM_install_dir\Doc\WhitePapers\`
- *UNIX:* `OV_DOC/WhitePapers/`

This information explains how to ensure that you do not break existing implementations.

You *might* need to install the white papers on Windows-based NNM management stations. See “White Papers” on page 57.

---

You can modify the `dedup.conf` file to change the list of affected alarms. For example, if you have multiple, identical alarms that you want to nest under the most recent instance of the alarm, do the following:

1. Open the `dedup.conf` file from the following location:
  - *Windows:* `install_dir\conf\dedup.conf`
  - *UNIX:* `OV_CONF/dedup.conf`
2. Make the following changes to increase de-duplication:
  - Add the SNMP MIB Object ID of the alarm.
  - Add the event SNMP-agent/IP-address source pair (optional).

## De-Duplication of Alarms

- Add the specific SNMP variable-binding (optional). See “What is an SNMP Variable-Binding and How Do I Identify One?” on page 387 for more information about SNMP variable-bindings.

See the *dedup.conf* reference page in NNM’s online help (or the UNIX manpage) for more information about how to edit the *dedup.conf* file.

---

### NOTE

If you need to disable the de-duplication feature, remove the # from the following line of the *dedup.conf* file.

```
#DEDUPLICATION=OFF
```

---

To deploy any changes you made to the *dedup.conf* file, you must restart the *ovalarmsrv* process. This causes NNM to read the contents of the *dedup.conf* file. To restart the *ovalarmsrv* process, use the following procedure:

1. Stop the *ovalarmsrv* process.

- *Windows*: As administrator, at the command line, type: `ovstop ovalarmsrv`
- *UNIX*: As root, at the command line, type: `ovstop ovlarmsrv`

2. Start the *ovalarmsrv* process.

- *Windows*: As administrator, at the command line, type: `ovstart ovalarmsrv`
- *UNIX*: As root, at the command line, type: `ovstart ovlarmsrv`

After you complete this process, alarms that were already posted to the Alarm Browser before you implemented the de-duplication are not removed. The de-duplication feature nests all new matching alarms as they arrive. The *newest* de-duplicated alarm always becomes the parent alarm, visible in the Alarm Browser.

## ECS Correlations

### NNM's Built-In Correlations

NNM includes a set of event correlations (the Bundled Edition correlations):

- **Connector Down Correlation System (page 350)**  
This feature encompasses more than event correlation. NNM's map processes and alarm monitoring processes have been enhanced to help you speed up the task of identifying the source of a problem on your network as it occurs.
- **ManageX Server Down Correlation (page 354)**  
When certain ManageX services become unavailable, this correlation combines HP OpenView ManageX events with NNM's topology events, creates new high level ManageX messages, and forwards them to the ManageX message reader. These messages explain why ManageX services are unavailable. Events sent to NNM by ManageX appear in the NNM Alarm Browser. Enable this correlation only if ManageX and NNM will be used together.
- **Pair Wise Correlation (page 356)**  
This correlation matches a *parent* event to one or more previously occurring *child* events; for example, an Interface Down alarm (OV\_IF\_Down) followed by an Interface Up alarm (OV\_IF\_Up) within a designated time frame. You can configure the behavior of alarms passing through this correlation.
- **Repeated Event Correlation (page 360)**  
This correlation identifies multiple duplicate alarms which are associated with a single physical event and bundles them together under a single alarm.
- **Scheduled Maintenance Correlation (page 364)**  
When conducting planned network maintenance that requires multiple devices to be off-line for a certain amount of time, this correlation allows you to exclude events generated because of the maintenance activity.

### Connector Down Correlation

Use of this correlation can prevent event storms when a router (or other connector device) goes down. In the event of a network failure, NNM automatically determines:

- Which device is malfunctioning.
- Which other network devices are impacted by this failure. That is, which functional network devices are now *inaccessible to the management station* over the network because of the failing device. These inaccessible devices are referred to as *secondary failures*.
- Which *inaccessible* network devices are *important* to the productivity of the organization and thus should be given high priority.

**Behavior** NNM automatically determines the route to a node that is down and checks each connector device along the path to that node to determine the primary malfunction.

Only the alarm about the primary device is logged to the Alarm Browser, making it easy for you to determine the root cause of the problem. To view the error messages regarding devices considered secondary failures:

- From the Alarm Browser, select the correlated alarm's message and select `Actions:Show Correlated Events` or double-click on the correlated column.
- From the web-based Alarm Browser, select the correlated alarm's message and select `Actions:Alarm Details` and select the `Correlations` tab.

NNM slows down polling traffic to all affected devices (i.e. devices considered secondary failures other than the primary malfunction) until the primary device is recovered. The map automatically updates to indicate the primary failure and the affected secondary failures below the primary failure.

You can establish a list of mission-critical devices (Important Node filter) that are never considered secondary failures.

**Setting Parameters** There are several configuration parameters for the `ConnectorDown` correlation contained within the correlation, itself. However, the `ConnectorDown` correlation is mainly controlled by NNM rather than the Event Correlation Service. It is recommended that you do not modify the correlation itself.

To set parameters for the `ConnectorDown` correlation, from any NNM submap, select `Options:Network Polling Configuration`. (See “Secondary Failure Polling” on page 177 for instructions.)

Make sure that the `Secondary Failures Polling Options` field is enabled because this field determines whether the `ConnectorDown` correlation is being used. The settings in the other fields are up to you (see the following examples for information about the behavior of different choices).

- **Important Node Filter**

Designate the name of the filter you created which lists all nodes that are mission-critical and should, therefore, never be placed in secondary-failure status. In the case of a failure resulting in devices contained in the list becoming inaccessible, alarm messages pertaining to these devices would continue to appear in the Alarm Browser window. (See “Router/Switch Health” on page 377 for information about the correlation of alarms from nodes listed in the Important Nodes filter.)

This filter resides in the same file that contains any discovery filters, DHCP filters, etc.:

— *Windows*: `install_dir\conf\C\filters`

— *UNIX*: `$OV_CONF/C/filters`

See the *OVfilterIntro* reference page (or the UNIX manpage) in NNM’s online help for more information about defining filters. See also *A Guide to Scalability and Distribution* for information about creating filters.

- **Important Node Behavior**

Designate which status change you prefer to receive for the devices listed in the important-node filter: `Down` (red on map) or `Unknown` (blue on map).

- **Secondary Node Polling Multiplier**

To prevent the generation of unnecessary polling traffic, specify a multiplier by which any scheduled polling should be extended during the failure for devices designated as secondary devices.

- **Secondary Node Behavior**  
Designate which status change you prefer to receive for devices determined to be secondary failures: Down (red on the map), Unknown (blue on the map), or Unchanged (no change reflected on the map).
- **Secondary Failure Alarm Suppression**  
Designate whether or not you want to prevent alarm messages about secondary failures from being posted to the Alarm Browser's main categories. If this function is selected, the secondary devices' alarm messages are displayed by selecting the primary alarm message in the Alarm Browser window and selecting the `Actions:Show Correlated Events` menu item.

---

**NOTE**

If you are using NNM Advanced Edition, see *A Guide to Scalability and Distribution* for information about configuring this feature on collection stations and their corresponding management station.

---

The following examples illustrate how these settings affect the behavior of NNM.

**Example 11-1      Example 1 Default Settings without Using a Filter**

Important Node Filter File: None  
Important Node's Failure Status: Down (irrelevant because no devices are identified in the filter)  
Secondary Node Failure Status: Unknown (blue on the map)  
Suppress Alarms for Secondary Failures: True  
Secondary Node Polling Multiplier: 2

During a network failure:

- The node determined to be the primary cause of the failure appears as a red (critical) symbol on the map, an alarm message is posted in the Alarm Browser, and polling intervals remain constant.
- Since no filter has been specified, no secondary nodes are considered important.
- All secondary-failure nodes are displayed as blue symbols (unknown) on the map. The alarm messages for all secondary devices do not appear in the Alarm Browser list. To view secondary alarm



messages, select the corresponding primary alarm and select the Actions:Show Correlated Events menu item. Once a node is identified as a secondary failure, its polling intervals are doubled.

### **Example 11-2      Example 2 Default Settings with a Filter**

```
Important Node Filter File: YourFilterName
Important Node's Failure Status: Down (red on the map)
Secondary Node Failure Status: Unknown (blue on the map)
Suppress Alarms for Secondary Failures: True
Secondary Node Polling Multiplier: 2
```

During a network failure:

- The device determined to be the primary cause of the failure appears as a red (critical) symbol on the map, an alarm message is posted in the Alarm Browser, and polling intervals remain constant.
- Any affected device that is specified as an important device in your filter file appears as a red symbol (critical) on the map, polling intervals remain constant, and alarm messages are posted in the Alarm Browser window.
- All secondary-failure interfaces on all nodes are displayed as blue symbols (unknown) on the map. The alarm messages for all secondary devices do not appear in the Alarm Browser list. To view secondary alarm messages, select the corresponding primary alarm and select the Actions:Show Correlated Events menu item. Once a node is identified as a secondary failure, its polling intervals are doubled.

### **Example 11-3      Example 3 Minimum Network Traffic During Event Correlation**

```
Important Node Filter File: YourFilterName
Important Node Behavior: Unknown (blue on the map)
Secondary Node Behavior: Ignore (no change on the map)
Secondary Failure Alarm Suppression: True
Secondary Node Polling Multiplier: 10
```

During a network failure:

- The device determined to be the primary cause of the failure appears as a red (critical) symbols on the map, an alarm message is posted in the Alarm Browser, and polling intervals remain constant.

## ECS Correlations

- Any affected device that is specified as an important device in your filter file appears as a blue symbol (unknown) on the map, polling intervals remain constant, and alarm messages are posted in the Alarm Browser window.
- All secondary-failure interfaces on all nodes are ignored, no change is reflected on the map. No alarm messages are generated for secondary devices. Once a node is identified as a secondary failure, its polling intervals are multiplied by 10.

### Example 11-4 Example 4 Behavior Like Previous Releases of NNM

To disable the *secondary device* strategy and configure NNM to consider all devices as primary, deselect the Secondary Failures Polling Options field to disable the Connector Down correlation. This will have the following effect.

```
Important Node Filter File: None
Important Node Behavior: (irrelevant because no devices identified in the filter)
Secondary Node Behavior: Down (red on the map)
Secondary Failure Alarm Suppression: False (all alarms posted)
Secondary Node Polling Multiplier: 1 (no change)
```

During a network failure:

- The device determined to be the primary cause of the failure appears as a red (critical) symbol on the map, an alarm message is posted in the Alarm Browser, and polling intervals remain constant.
- Since no filter has been specified, no secondary nodes are singled out as important nodes.
- All secondary-failure interfaces on all nodes are displayed as red symbols (critical) on the map. The alarm messages for all secondary devices are sent directly to the Alarm Browser list. No polling intervals are changed.

### ManageX Server Down Correlation

The `MgXServerDown` correlation analyzes the network health, generates new high level ManageX messages, and delivers these messages to the ManageX message reader. The resulting messages explain why certain ManageX services are unavailable. ManageX messages also appear in the NNM Alarm Browser once this correlation is configured and enabled.

The message information is of value to the ManageX system administrator as it explains what portions of the network are interfering with ManageX operation. This information eliminates the need for the ManageX administrator to contact the network administrator and improves the efficiency of both the network administrator and the system administrator.

**Behavior** ManageX console events are picked up by the *NNM Lights Out policy* (nnmIntegration.mxc) and converted to an SNMP trap which is sent to NNM. These events route through the ECS `MgXServerDown` correlation for evaluation by NNM.

If NNM discovers that a ManageX device is down, the ECS `MgXServerDown` correlation will determine if the interface-down event is due to a problem on a connector device (router interface, switch, etc.). The result of the analysis is encapsulated in an event that NNM converts into an SNMP trap and sends to the ManageX NNM WMI policy (nnmWmiEventPolicy.mxc) where it is converted into a ManageX console message for display in the ManageX message reader.

**Setting Parameters** There are numerous configurable parameters contained in the `MgXServerDown` correlation. Complete the following steps if you want to review parameter definitions or modify parameters contained within the `MgXServerDown` correlation.

---

**TIP**

There are several ways to access the event correlation features. For more information, from any submap, select `Tools:HP OpenView Launcher`. Select the `[?]` tab. Click `Tasks, Event Correlation Management`. Read the information under *Accessing the Event Correlation Configuration Windows*.

- 
1. From any submap, select `Options:Event Configuration`. This launches the `Event Configuration` window.
  2. From NNM's `Event Configuration` window, select `Edit:Event Correlation`. This brings up the `ECS Configuration` window.
  3. From the `ECS Configuration` window, highlight the `MgXServerDown` correlation and select `Describe` to review the correlation description.

4. From the ECS Configuration window, highlight the MgXServerDown correlation and select Modify to bring up the MgXServerDown-Modify screen.
5. To view parameter descriptions or modify any of the configurable parameters, select the parameter, then select View/Modify.

---

**TIP**

From the View/Modify window, select Help for general information on making correlation modifications.

---

### Pair Wise Correlation

This correlation matches a *parent* event to one or more previously occurring *child* events. You can configure child events to:

- Display in the Alarm Browser while awaiting arrival of a parent event, or
- Only display in the Alarm Browser if the specified time window is exceeded without the arrival of a matching parent event,
- Upon arrival of a parent event, be removed from the Alarm Browser and nested beneath the parent event, or set to Acknowledged.

For example, consider several repeated OV\_IF\_Down events followed by an OV\_IF\_Up event within a designated time frame. This correlation could be configured to either remove or acknowledge the OV\_IF\_Down events upon arrival of the OV\_IF\_Up event. The OV\_IF\_Down events are nested under the OV\_IF\_Up alarm message in the Alarm Browser.

You can add SNMP traps and events to the PairWise correlation, provided that you know how the source that is sending the trap is identified. The SNMP MIB Object ID (OID) works in combination with one of the following to specify the source. The combination being used is determined by the developer of the underlying MIB. You must know the appropriate combination in order to configure additional events into this correlation:

- agent-addr attribute
- Any source wildcard (your choice)
- SNMP variable-binding name
- SNMP variable-binding position

**Behavior** You control the behavior of this correlation. The following three examples explain some of your choices. Remember that in order for an event to be displayed within the Alarm Browser, the event's configuration must *not* be set to:

- Don't log or display
- Log only

See "Event Configuration Overview" on page 404 for information about setting an event's configuration.

### **Example 11-5 Don't Display the Child Events or the Parent Event**

```
ChildEventImmediateOutput = False  
InhibitParentOfInhibitedChild = True
```

The correlation monitors the defined event parent/child relationships in the background. You never see the parent or any child events in the Alarm Browser unless the time of the defined window period is exceeded and no parent event arrives. If no parent is received, the child events are displayed in the Alarm Browser.

### **Example 11-6 Display the Parent Event, Nest Child Events Beneath Parent**

```
ChildEventImmediateOutput = False  
InhibitParentOfInhibitedChild = False
```

The correlation monitors the defined event parent/child relationships in the background. When the parent event is received, the parent event is displayed in the Alarm Browser with the child events nested beneath. If no parent is received within the defined window period, the child events are displayed in the Alarm Browser.

### **Example 11-7 Display Both the Parent and Child Events in the Alarm Browser**

```
ChildEventImmediateOutput = True  
InhibitParentOfInhibitedChild = ignored (True or False)
```

Child events are displayed in the Alarm Browser upon arrival. The correlation monitors the defined event parent/child relationships, and when the parent event is received, the parent event is displayed in the Alarm Browser with the child events nested beneath. You choose whether the child events are removed from the Alarm Browser list or set to Acknowledged upon arrival of the parent event.

To view nested child alarms:

## ECS Correlations

- From the Alarm Browser, select the parent alarm message and select `Actions:Show Correlated Events` or double-click on the correlated column.
- From the web-based Alarm Browser, select the parent alarm message and select `Actions:Alarm Details` then select the `Correlations` tab. You can also select the parent alarm message, click on the view button and then select the `Correlations` tab.

**Setting Parameters** Complete the following steps if you want to review parameter definitions or modify parameters contained within the PairWise correlation.

---

### TIP

There are several ways to access the event correlation features. For more information, from any submap, select `Tools:HP OpenView Launcher`. Select the [?] tab. Click `Tasks, Event Correlation Management`. Read the information under *Accessing the Event Correlation Configuration Windows*.

1. From any submap, select `Options:Event Configuration`. This launches the Event Configuration window.
2. From NNM's Event Configuration window, select `Edit:Event Correlation`. This brings up the ECS Configuration window.
3. From the ECS Configuration window, highlight the PairWise correlation and select `Describe` to review the correlation description.
4. From the ECS Configuration window, highlight the PairWise correlation and select `Modify` to bring up the PairWise-Modify screen.
5. To view parameter descriptions or modify any of the configurable parameters, select the parameter, then select `View/Modify`.

---

### TIP

From the `View/Modify` window, select `Help` for general information on making correlation modifications.

---

This correlation allows configuration of the following parameters.

These two tables define which events are being monitored by the correlation and define the parent/child relationships:

- The `InputEventTypeListStringSources` table is used when the event source is a simple string which is not an IP address. This table is used to set event types for input into the correlation and determine the parent and child event sources.
- The `InputEventTypeList` table is used when the event source is an IP address. This table is used to set event types for input into the correlation and determine the parent and child event sources.

The following two parameters control when alarms appear in the Alarm Browser:

- The `ChildEventImmediateOutput` parameter determines whether the `PairWise` correlation outputs child events immediately or if it holds onto child events for further processing and possible cancellation by a parent event.

`False` = The `PairWise` correlation holds onto child events for the duration of the `PairedTimeWindow`. If the `PairWise` correlation receives a matching parent event from the same source, within the appropriate amount of time, the child event is nested beneath the parent event in the Alarm Browser. If the `PairWise` correlation does not receive a matching parent event within the `PairedTimeWindow`, the child event is displayed in the Alarm Browser.

`True` = Child events are displayed in the Alarm Browser upon arrival. Child events may be deleted or acknowledged in the Alarm Browser at a later time if a matching parent event arrives.

- The `InhibitParentOfInhibitedChild` parameter determines whether parent events are displayed in the Alarm Browser.

---

**NOTE**

If the `ChildEventImmediateOutput` is `true`, the `PairWise` correlation ignores the `InhibitParentOfInhibitedChild` parameter and sends parent events to the Alarms Browser immediately.

---

`True` = The correlation monitors the defined event parent/child relationships in the background. You never see either parent or child alarms in the Alarm Browser unless the time of the defined window period is exceeded and no parent event arrives. If no parent is received, the child events are displayed in the Alarm Browser.

`False` = The parent event is displayed in the Alarm Browser upon arrival, with all child events nested beneath.

The following parameters fine tune the correlation:

- The `DeleteOrAcknowledge` parameter determines whether a child event is deleted or acknowledged from the Alarm Browser once it is matched to a parent event (assuming that `ChildEventImmediateOutput = True`).
- The `PairedTimeWindow` is the maximum time that the child event waits for a parent event. However, the `PairWise` correlation has a built in *pattern delete* feature. If a parent event arrives outside of the defined window, the Alarm Browser is scanned for any defined child events from the same source as the parent event.
- The `IgnoreSecondaryFailureEvents` parameter determines whether this correlation monitors events from devices identified as secondary failures. When set to `True`, secondary failures are monitored by the `ConnectorDown` correlation (page 350), and not by the `PairWise` correlation.

### Repeated Event Correlation

This correlation allows multiple alarms which are associated with a single physical event to be bundled together and replaced by a single alarm in NNM's Alarm Browser.

For example, the OpenView `OV_PhysAddr_Mismatch` alarm is one of several alarms that are configured and active by default. When the physical address of a device changes, NNM generates multiple events as neighboring systems discover the change and NNM reads this conflicting information. The `RepeatedEvent` correlation suppresses the multiple instances of the physical address change alarms and correlates them into one alarm for any particular node. You specify the time interval; for example, after receipt of the first of many repeated alarms, suppress duplicates for 2 hours. Then generate another alarm and resume suppression of duplicate alarms for two more hours.



You can add other SNMP traps to the `RepeatedEvent` correlation for suppression of duplicate alarms, provided that you know the following information. Event Identifiers (MIB OIDs) work in combination with one of the following to specify how to obtain the source of the incoming trap:

- `agent-addr` attribute
- Any source wildcard (your choice)
- SNMP variable-binding name
- SNMP variable-binding position

The combination being used is determined by the developer of the underlying MIB. You must know the appropriate combination in order to configure additional events into this correlation.

**Behavior** In NNM's Alarm Browser, all traps with the same key identifier [OID +source specifier] are nested under the first trap received for the specified time period. Only the first alarm in the series is posted into the Alarm Browser's main list. To view the rest of the alarms:

- From the Alarm Browser, select the primary message and select `Actions: Show Correlated Events` or double-click on the correlated column.
- From the web-based Alarm Browser, select the primary message and select `Actions: Alarm Details`, then select the `Correlations` tab. You can also select the primary message, click on the view button and then select the `Correlations` tab.

**Setting Parameters** Complete the following steps if you want to review parameter definitions or modify parameters contained within the `RepeatedEvent` correlation.

---

**TIP**

There are several ways to access the event correlation features. For more information, from any submap, select `Tools:HP OpenView Launcher`. Select the [?] tab. Click `Tasks, Event Correlation Management`. Read the information under *Accessing the Event Correlation Configuration Windows*.

1. From any submap, select `Options:Event Configuration`. This launches the `Event Configuration` window.

## ECS Correlations

2. From NNM's Event Configuration window, select **Edit: Event Correlation**. This brings up the ECS Configuration window.
3. From the ECS Configuration window, highlight the **RepeatedEvent correlation** and select **Describe** to review the correlation description.
4. From the ECS Configuration window, highlight the **RepeatedEvent correlation** and select **Modify** to bring up the **RepeatedEvent-Modify** screen.
5. To view parameter descriptions or modify any of the configurable parameters, select the parameter, then select **View/Modify**.

---

### TIP

From the **View/Modify** window, select **Help** for general information on making correlation modifications.

---

The **RepeatedEvent** correlation contains the following parameters:

- The *HoldFirstEventTime* parameter sets an overall maximum time that initial events are held for later correlation with future events.
- The *MaxHoldFirstEvents* parameter sets an overall maximum number of first events to hold for later correlation with future events.
- The *InputEventTypeList* is a table that specifies the event types and event source identifiers that are input to this correlation. The following **Generic Trap Strings** and **Input Event OID** descriptions are included as event types in the *InputEventTypeList* table.

**Generic Trap String:** The following generic trap names are contained within the correlation. Valid generic trap names are:

`coldStart`

`warmStart`

`linkDown`

`linkUp`

`authenticationFailure`

`egpNeighborLoss`

By default, all of the generic traps are set to false (event not enabled) and are not evaluated by the `RepeatedEvent` correlation.

**Input Event OID: The Specific Event Types (Input Event OID)** specify the trap OID. Two Input Event OIDs are set to true (by default):

`OV_PhysAddr_Mismatch (1.3.6.1.4.1.11.2.17.1.0.58982401)`

`OV_Node_Added (.1.3.6.1.4.1.11.2.17.1.0.58785794)`

Enabling a particular row in the `InputEventTypeList-Modify` screen allows the event to be used as an input event into the correlation.

---

**TIP**

Make sure that the Event Configuration settings for the trap you are enabling is *not* set to either of the following. If it is, your correlated alarm will never show up in the NNM Alarm Browser:

- Don't log or display
- Log only

See “Event Configuration Overview” on page 404.

- 
- The `RepeatedTimeWindow` parameter sets an overall maximum time to correlate duplicate events. This value is used for both fixed (`RollingWindow=false`) and rolling (`RollingWindow=true`) time windows.
  - The `CreateUpdateEvent` parameter allows you to specify whether or not to delete the original alarm from the Alarm Browser and create a new summary event when the time limit for this correlation is exceeded. If `CreateUpdateEvent` is true, the `RepeatedEvent` correlation creates a new, identical summary event that contains a count of the correlated events. If `CreateUpdateEvent` is false, the original alarm is not deleted from the Alarm Browser and no update event is created.
  - Setting the `RollingWindow` parameter to true means that the time window is not fixed to a set time duration. The rolling time window resets each time the correlation receives another duplicate event.

For example, the `RepeatedTimeWindow` parameter is set to 10 minutes with the `RollingWindow` parameter set to true. An event reports a conflicting MAC layer address. A second identical event arrives (repeats) within 10 minutes of the initial event. The time window for the correlation extends for another 10 minutes and the event is nested beneath the initial event in the Alarm Browser.

### Scheduled Maintenance Correlation

Computer nodes and network segments are occasionally scheduled for maintenance. Such an occurrence typically results in a number of SNMP traps or HP OpenView events being generated. The `ScheduledMaintenance` correlation prevents unnecessary alarms during maintenance.

**Behavior** When the `ScheduledMaintenance` correlation detects the first event from a specified node or network segment within the specified time range, an event is generated and posted in the NNM Alarm Browser with the message: `Scheduled Maintenance`. The triggering event and all subsequent matching events are nested beneath the `Scheduled Maintenance` alarm. To view the nested alarms:

- From the Alarm Browser, select the primary message and select `Actions:Show Correlated Events` or double-click on the correlated column.
- From the web-based Alarm Browser, select the primary message and select `Actions:Alarm Details` and select the *Correlations* tab.

**Setting Parameters** Complete the following steps if you want to configure and enable the `ScheduledMaintenance` Correlation.

---

**TIP**

There are several ways to access the event correlation features. For more information, from any submap, select `Tools:HP OpenView Launcher`. Select the [?] tab. Click `Tasks, Event Correlation Management`. Read the information under *Accessing the Event Correlation Configuration Windows*.

- 
1. From any submap, select `Options:Event Configuration`. This launches the Event Configuration window.

2. From NNM's Event Configuration window, select `Edit:Event Correlation`. This brings up the ECS Configuration window.
3. If you want to enable or disable the `ScheduledMaintenance` correlation: from the ECS Configuration window, highlight the `ScheduledMaintenance` correlation. Then select either `Enable` or `Disable`.
4. From the ECS Configuration window, highlight the `ScheduledMaintenance` correlation and select `Describe` to review the correlation description.
5. From the ECS Configuration window, highlight the `ScheduledMaintenance` correlation and select `Modify` to bring up the `ScheduledMaintenance-Modify` screen.
6. To view or modify the affected network devices, or set the name of the planned outage, or activate/deactivate a planned outage: select `MaintenanceList` from the `ScheduledMaintenance-Modify` window and click on `View/Modify`.
7. To view or modify the date, time, and duration of a planned outage: Select `OutageTimeSpecification` from the `ScheduledMaintenance-Modify` window and click on `View/Modify` to make changes. It's important to note that a time window name is specified and configured in the `OutageTimeSpecification-Modify` window. This time window name is then referenced in the `MaintenanceList` definition.
8. To view any other parameter descriptions or modify any of the other configurable parameters, select the parameter, then select `View/Modify`.

---

**TIP**

From the `View/Modify` window, select `Help` for general information on making correlation modifications.

---

## Command Line Control

You can enable, disable, and control correlations from the `Options:Event Configuration` or `Edit:Event Correlation` menu item. However, if you need to debug a test correlation or enable tracing or logging for support purposes, use the `ecsmgr` command line utility. Possible tasks include:

**ECS Correlations**

- Starting or stopping event logging.
- Configuring error logging and trace logging.
- Creating a new stream.

See the *ecsmgr* reference page in NNM's online help (or the UNIX manpage) for more information.

**Correlation File Structure**

Each correlation is comprised of four files. Correlations are stored in the following directory:

- *Windows*: `install_dir\conf\ecs\circuits\*.*`
- *UNIX*: `$OV_CONF/ecs/circuits/*.*`

Do not edit these files directly. Only the `*.ds` file can be altered by using the `Options:Event Configuration`, or `Edit:Event Correlation` menu items.

**Table 11-1**                      **Correlation File Structure**

<code>correlation.eco</code>	Compiled correlation file (not editable).
<code>correlation.ds</code>	Data Store file containing externally configurable parameter values (text).
<code>correlation.fs</code>	Optional Fact Store file containing topological or other information describing relationships (text).
<code>C/correlation.param</code> or <code>\$LANG/correlation.param</code>	Parameter file that describes the parameters configurable by the ECS Event Configuration window (text).

**Troubleshooting****Correlation Not Displayed**

If the correlation you want is not displayed:

- In the ECS Event Configuration window, click [Update View] to refresh the list display. Updates made by other users and updates made from the command line are not reflected in the list of correlations until you click [Update View].
- The correlation may not have been loaded (installed).

### **Correlation Status Incorrect**

If the Status of the correlation appears to be incorrect:

- In the ECS Event Configuration window, select the appropriate stream. Correlations are enabled in the selected stream only and the correlation status reflects the status for the selected stream only. A correlation may be enabled in one stream and disabled in others.
- Click [Update View] to refresh the list display.

### **ECS Event Configuration Window Hangs**

If the ECS Event Configuration window appear to hang when you click [Verify Table] or [OK] after modifying a parameter value:

- Wait until the server responds. Verification is performed on the server and can take some time if the server is very busy.
- Check that the connection with the server has not been severed.

### **Too Many Alarms Are Showing Up**

Did you change the NNM Event Configuration default settings for any events from "log-only" to display in the NNM Alarm Browser? Consider changing some of them back to "log-only" For example, you do not need to display the Link Down events received from your Cisco router because NNM is already monitoring status. The NNM OV\_IF\_Down event is the same as the hardware's Link Down trap.

Expand the configuration of de-duplication to include any repetitive events (see "De-Duplication of Alarms" on page 347).

### **Obtaining Additional Event Correlations**

NNM includes several **Bundled Edition** correlations. No additional license or software is required to use these correlations. HP tests and supports these correlations.

There are three additional sources for correlations that can run in NNM:

- **Contributed Edition** correlations are provided by HP OpenView Partners. Check the HP OpenView web page for links (also check OpenView partners' web sites for access to these correlations). No additional license or software is required. HP does not test or support the Contributed Edition correlations. They are tested and supported by the partner.
- **Extended Edition** correlations can be obtained from system integrators, such as HP Consulting Service (see “HP Consulting Service” on page 58) and others. They are tested and supported by the supplier.
- **Designer Edition** correlations are written by you. You must purchase and use the HP OpenView ECS Designer product. HP does not test or support the Designer Edition correlations. They must be tested and supported by you.

If you add correlations to NNM, they must be installed and their initial parameter values configured. When a correlation is enabled, its memory tables must fill with events before correlation becomes effective. Like the correlations supplied with NNM, you can then make further modifications to the parameter values while the correlation is enabled.

### So You Want To Develop Your Own Correlation?

Developing your own event correlations is a powerful option but a non-trivial task. You should consider carefully whether this is something you want to do or whether it might be quicker, simpler, and cheaper to contract with a specialist to supply the correlation or correlations you need.

If you need only one or two special correlations, then it is probably not worthwhile developing them yourself. Instead, you should consider contracting a specialist to develop them for you. Contact HP Consulting Service (see “HP Consulting Service” on page 58) or other system integrators for assistance.

To develop a correlation you need to:

- Purchase the HP OpenView ECS Designer product, a separately licensed component.
- Learn to develop correlations (ECS Designer training, and/or documentation).



- Design and debug the correlation, data store file, and parameter file.
- If required, develop processes to use the ECS Annotation and/or Event I/O APIs, as described in the *HP OpenView ECS Developer's Guide*.
- If you are using NNM Advanced Edition in a distributed environment, deploy the correlation to each management station and collection station, as described in the documentation that you received when you purchased ECS Designer.

**When to Design Your Own Event Correlations** There are several key factors which indicate that a new correlation is required, or that an existing correlation needs to be reconstructed:

- No existing correlation provides the correlation you need.
- Parameters of an existing correlation need to be set to values that are outside the minimum and maximum allowed in the ECS Event Configuration window.
- The correlation requires access to external data (for example, to look up an inventory database) and therefore needs to use ECS annotation.
- Special event I/O requirements exist.

### What is Involved

---

#### NOTE

All de-duplication configurations, correlators defined within Correlation Composer, and all correlations defined within ECS must *work well together* on your NNM management station. Before you begin, it is critical that you print out and read the following white paper:

Developing\_NNM\_Event\_Reduction.pdf

- *Windows:* `NNM_install_dir\Doc\WhitePapers\`
- *UNIX:* `$OV_DOC/WhitePapers/`

This information explains how to ensure that you do not break existing implementations.

You *might* need to install the white papers on Windows-based NNM management stations. See “White Papers” on page 57.

## ECS Correlations

To design an effective correlation, you must:

- Understand typical network event patterns and what they mean.
- Know the architecture of your network management system.
- Identify external data access (ECS Annotation) requirements.
- Define the problem concisely in terms of filtering and time-related if-then statements.
- Determine if all data is available from the events themselves. If not determine if:
  - The external data is subject to change or is relatively stable.
  - The required data is small enough to store in memory.
  - The access speed and reliability is sufficient when extracting from your database (particularly if the database is remote).

Then, to develop the correlations, in addition to purchasing the HP OpenView ECS Designer product, you need to:

- Set up a test platform.
- Obtain or develop sample logs of events (perhaps use NNM backup files).
- If Event I/O or Annotation is required, refer to the *HP OpenView ECS Developer's Guide* and obtain an ANSI C/C++ compiler.
- If ASCII events must be correlated, use the ASCII Module of the HP OpenView ECS Designer product.

## Correlation Composer Correlators

### NNM's Built-In Correlators

NNM includes a set of correlators. The correlators are logically divided into two namespaces: `OV_NNM_Basic` and `OV_NodeIf`.

The `OV_NNM_Basic` namespace contains the following correlators:

- “Cisco Chassis Failure” on page 372.  
Monitors Cisco traps for three error conditions: temperature, fan failure, and power supply fault. Generates a new alarm if the condition persists for the specified time period.
- “Multiple Reboots on Routers/Switches” on page 373.  
Listens for `coldStart` and `warmStart` traps and creates a new alarm when more than  $N$  `coldStart`/`warmStart` traps are received within  $M$  minutes from a specific SNMP-agent/IP-address pair (defaults  $N=4$ ,  $M=5\text{min}$ ).
- “Router/Switch Intermittent Status Changes” on page 375.  
Listens for `OV_IF_Down` alarms from routers and switches, and creates a new alarm when more than  $N$  interface-down events are received within  $M$  minutes from a specific interface (defaults  $N=5$ ,  $M=30\text{min}$ ). Only the new `OV_Intermittent` alarm appears in the NNM Alarm Browser.

The `OV_NodeIf` namespace contains the following correlators:

- “Router/Switch Health” on page 377.  
(actually a group of three correlators) Correlates Interface status alarms with their related router or switch Node status alarm. `OV_IF_Unknown` and `OV_IF_Down` status alarms from interfaces within routers or switches are suppressed and nested beneath the Node-status alarm.

In addition to the built-in correlators provided with NNM, you can construct your own correlators with Correlation Composer. For more information about creating correlators, see “Creating Additional Correlators” on page 382.

### Cisco Chassis Failure

Correlates events forwarded from switches supporting the CISCO-STACK-MIB.my MIB.

**Behavior** Certain Cisco and HP routers and switches that support the CISCO-STACK MIB can be configured to send traps that report chassis error conditions to the NNM management station. This correlator monitors the SNMP variable-binding data within the Cisco traps that provides information about the reasons for a failure.

If a *temperature, fan failure, or power supply* fault condition persists for a period longer than the configured time interval, one of the following events is generated and sent to the NNM Alarm Browser:

- OV\_Chassis\_Temperature (.1.3.6.1.4.1.11.2.17.1.0.58982424)
- OV\_Chassis\_FanFailure (.1.3.6.1.4.1.11.2.17.1.0.58982425)
- OV\_Chassis\_PowerSupply (.1.3.6.1.4.1.11.2.17.1.0.58982426)

Conditions that persist for a time less than the specified Window Period are discarded.

**Setting Parameters** Complete the following steps if you want to review parameter definitions or modify parameters contained within the OV\_Chassis\_Cisco correlator.

---

#### TIP

There are several ways to access the event correlation features. For more information, from any submap, select Tools:HP OpenView Launcher. Select the [?] tab. Click Tasks, Event Correlation Management. Read the information under *Accessing the Event Correlation Configuration Windows*.

- 
1. From any submap, select Options:Event Configuration. This launches the Event Configuration window.
  2. From NNM's Event Configuration window, select Edit:Event Correlation. This brings up the ECS Configuration window.
  3. From the ECS Configuration window, select the 'default' stream. Then, highlight Composer in the correlation table and select Modify. The Correlation Composer window appears in your web browser.

4. In the Correlation Composer window, select the `OV_NNM_Basic` namespace from the NameSpace table. Its correlators are displayed in the Correlator Store.
  5. Double-click `OV_Chassis_Cisco` to display the Description tab.
  6. Carefully read the information in the Description tab. The only configurable parameter, by default, is:
    - Window Period
- If you need to modify values of other parameters, open Correlation Composer in developer mode. See *HP OpenView Correlation Composer's Guide* for more information about Correlation Composer in developer mode.
7. Click the Definition tab to access the configurable parameter setting. Click [Help] for information about each field.
  8. After making the desired change, click [OK] and close the correlator configuration window and return to the Correlation Composer main window.
  9. Save your changes by clicking `File:Save`. This updates the correlator fact store file associated with `OV_NNM_Basic`.
  10. To activate your changes, click `File:Close` and then click `Correlations:Deploy`.
  11. Exit the Correlation Composer main window.

### Multiple Reboots on Routers/Switches

The Multiple Reboots correlator detects and reports problems with routers and switches that reboot multiple times within a specified time window.

**Behavior** The ECS PairWise correlation (page 356) suppresses `OV_Node_Down` alarms when the corresponding `OV_Node_Up` alarm arrives. Therefore, you may never see these events in the NNM Alarm Browser. This `OV_MultipleReboots` correlator detects a repetitive down/up situation within routers and switches and generates an `OV_Multiple_Reboots` alarm to warn you of a potential problem.

If a `coldStart` or `warmStart` trap arrives and no such traps have previously been seen for this SNMP-agent/IP-address pair, a new interval is started. If `coldStart` or `warmStart` traps already exist for this system, a counter is updated and checked to see if the number of

## Correlation Composer Correlators

counts exceeds the configured threshold. If the count exceeds the threshold and the trap is still within the current interval for this system, an `OV_Multiple_Reboots` alarm is posted in the NNM Alarm Browser. All further `coldStart` or `warmStart` traps within the specified time interval are nested under the `OV_Multiple_Reboot` alarm.

Once the interval expires, the sequence begins again.

**Setting Parameters** Complete the following steps if you want to review parameter definitions or modify parameters contained within the `OV_MultipleReboots` correlator.

---

### TIP

There are several ways to access the event correlation features. For more information, from any submap, select `Tools:HP OpenView Launcher`. Select the `[?]` tab. Click `Tasks, Event Correlation Management`. Read the information under *Accessing the Event Correlation Configuration Windows*.

- 
1. From any submap, select `Options:Event Configuration`. This launches the Event Configuration window.
  2. From NNM's Event Configuration window, select `Edit:Event Correlation`. This brings up the ECS Configuration window.
  3. From the ECS Configuration window, select the 'default' stream. Then, highlight `Composer` in the correlation table and select `Modify`. The Correlation Composer window appears in your web browser.
  4. In the Correlation Composer window, select the `OV_NNM_Basic` namespace from the `NameSpace` table. Its correlators are displayed in the Correlator Store.
  5. Double-click the `OV_MultipleReboots` correlator to display the Description tab.
  6. Carefully read the information in the Description tab. The only configurable parameters, by default, are:
    - Count
    - Window Period

If you need to modify values of other parameters, open Correlation Composer in developer mode. See *HP OpenView Correlation Composer's Guide* for more information about Correlation Composer in developer mode.

7. Click the `Definition` tab to access the configurable parameter settings. Click `[Help]` for information about each field.
8. After making the desired changes, click `[OK]` and close the correlator configuration window and return to the Correlation Composer main window.
9. Save your changes by clicking `File:Save`. This updates the correlator fact store file associated with `OV_NNM_Basic`.
10. To activate your changes, click `File:Close` and then click `Correlations:Deploy`.
11. Exit the Correlation Composer main window.

### Router/Switch Intermittent Status Changes

Identifies routers or switches that are reporting intermittent up/down status.

**Behavior** If an interface is continuously going down and then coming up, the ECS `PairWise` correlation (page 356) cancels the `OV_IF_Down` event when the `OV_IF_Up` event arrives. Therefore, you would never see these events in the NNM Alarm Browser.

This correlator (`OV_Connector_IntermittentStatus`) detects a repetitive interface down/up situation within a router or switch and generates an `OV_Intermittent` alarm if the `OV_IF_Down` event occurs the specified number of times (`Count`) during the specified time (`Window Period`).

**Setting Parameters** Complete the following steps if you want to review parameter definitions or modify parameters contained within the `OV_Connector_IntermittentStatus` correlator.

---

**TIP**

There are several ways to access the event correlation features. For more information, from any submap, select `Tools:HP OpenView Launcher`. Select the `[?]` tab. Click `Tasks, Event Correlation Management`. Read the information under *Accessing the Event Correlation Configuration Windows*.

---

1. From any submap, select `Options:Event Configuration`. This launches the `Event Configuration window`.
2. From NNM's `Event Configuration window`, select `Edit:Event Correlation`. This brings up the `ECS Configuration window`.
3. From the `ECS Configuration window`, select the 'default' stream. Then, highlight `Composer` in the correlation table and select `Modify`. The `Correlation Composer window` appears in your web browser.
4. In the `Correlation Composer window`, select the `OV_NNM_Basic` namespace from the `NameSpace table`. Its correlators are displayed in the `Correlator Store`.
5. Double-click the `OV_Connector_IntermittentStatus` correlator to display the `Description tab`.
6. Carefully read the information in the `Description tab`. The only configurable parameters, by default, are:
  - Count
  - Window Period

If you need to modify values of other parameters, open `Correlation Composer` in developer mode. See *HP OpenView Correlation Composer's Guide* for more information about `Correlation Composer` in developer mode.

7. Click the `Definition tab` to access the configurable parameter settings. Click `[Help]` for information about each field.
8. After making the desired changes, click `[OK]` and close the correlator configuration window and return to the `Correlation Composer main window`.
9. Save your changes by clicking `File:Save`. This updates the correlator fact store file associated with `OV_NNM_Basic`.



10. To activate your changes, click `File:Close` and then click `Correlations:Deploy`.
11. Exit the Correlation Composer main window.

### Router/Switch Health

Suppresses interface status alarms from devices other than routers or switches so that these alarms never appear in the NNM Alarm Browser. The assumption is that you do not wish to see end-node interface status alarms.

Displays, in the NNM Alarm Browser, the important interface status alarms from your routers and switches. Suppresses the unimportant interface status alarms from those same routers and switches.

Router/switch health is actually determined by a group of three correlators:

- `OV_NodeIf_NodeDown`
- `OV_NodeIf_NodeNotConnector`
- `OV_NodeIF_PrimaryIFUnknown`

**Behavior** Suppresses interface status alarms from devices other than routers or switches.

Suppresses interface status alarms from unused (unconnected) ports within routers or switches.

Addresses the following failure scenarios that the `ECS ConnectorDown` correlation (page 350) doesn't detect:

- If fewer than *all* the interfaces on a router or switch fail, the interface status alarms display in your Alarm Browser. Intermediate node status alarms (`OV_Node_Warning`, `OV_Node_Marginal`, or `OV_Node_Major`) are configured as *log-only*, so they never appear in your Alarm Browser.
- If *all* the interfaces on a router or switch fail, `OV_Node_Down` alarm displays in your Alarm Browser with the interface alarms nested underneath.

- If a router or switch failure is a secondary failure (for example a Switch B is inaccessible due to a Switch A failure), then the ECS ConnectorDown correlation suppresses all of the events associated with Switch B. The suppressed Interface alarms are nested under the primary Switch A OV\_Node\_Down alarm.

However, if the Secondary Failure Polling configuration identifies Switch B as an **Important Node** (listed in your Important Node filter, see page 350 and page 177) the Switch B secondary failure events are not suppressed by the ECS ConnectorDown correlation. In this case, the Switch B OV\_Node\_Unknown alarm is displayed in the NNM Alarm Browser. The OV\_NodeIf group of correlators nest the associated Interface status alarms beneath the parent node alarm. The Switch B OV\_Node\_Unknown alarm is not nested under the Switch A alarm.

**Setting Parameters** Complete the following steps if you want to review parameter definitions or modify parameters contained within the OV\_IF\_NodeDown correlator.

---

**TIP**

There are several ways to access the event correlation features. For more information, from any submap, select Tools:HP OpenView Launcher. Select the [?] tab. Click Tasks, Event Correlation Management. Read the information under *Accessing the Event Correlation Configuration Windows*.

1. From any submap, select Options:Event Configuration. This launches the Event Configuration window.
2. From NNM's Event Configuration window, select Edit:Event Correlation. This brings up the ECS Configuration window.
3. From the ECS Configuration window, select the 'default' stream. Then, highlight Composer in the correlation table and select Modify. The Correlation Composer window appears in your web browser.
4. In the Correlation Composer window, select the OV\_NodeIf namespace from the NameSpace table. Its correlators are displayed in the Correlator Store.
5. Double-click the OV\_IF\_NodeDown correlator to display the Description tab.

6. Carefully read the information in the `Description` tab. The only configurable parameter, by default, is:
    - `Window Period`
- If you need to modify values of other parameters, open `Correlation Composer` in developer mode. See *HP OpenView Correlation Composer's Guide* for more information about `Correlation Composer` in developer mode.
7. Click the `Definition` tab to access the configurable parameter setting. Click `[Help]` for information about each field.
  8. After making the desired change, click `[OK]` and close the correlator configuration window and return to the `Correlation Composer` main window.
  9. Save your change by clicking `File:Save`. This updates the correlator fact store file associated with `OV_NodeIf`.
  10. To activate your change, click `File:Close` and then click `Correlations:Deploy`.
  11. Exit the `Correlation Composer` main window.

**Disabling the `OV_NodeIF` Correlator Group** In releases prior to NNM 6.31, *node-status-events* were displayed in the Alarm Browser, and *interface-status-events* were configured as log-only so that they never displayed in the Alarm Browser. In this release of NNM, interface status events are usually displayed unless a catastrophic failure is detected (all interfaces within a node are down). If a catastrophic event is detected, the node alarm is displayed with the interface alarms nested beneath.

This procedure explains how to return to the old NNM behavior, if desired:

1. From any submap, select `Options:Event Configuration`. This launches the `Event Configuration` window.
2. Select `OpenView`.
3. Sort the events using the `View:Sort->Sort by Event Name` command.
4. Edit the `OV_IF_Unknown` and `OV_IF_Down` events and change the `Category` field to `Log only`.
5. Edit the following events and change the `Category` field to `Status Alarms` instead of `Log only`:

## Correlation Composer Correlators

- OV\_Network\_Major
- OV\_Network\_Critical
- OV\_Segment\_Major
- OV\_Segment\_Critical
- OV\_Node\_Warning
- OV\_Node\_Marginal
- OV\_Node\_Major
- OV\_Remote\_Mgr\_Up

6. Select [OK].

7. Save your changes using the File:Save command.

8. Exit using the File:Exit command.

Once you have the events reconfigured, complete the following to disable the group of three OV\_NodeIF correlators:

1. From any NNM submap, select Tools:HP OpenView Launcher. Select the [?] tab. Click:

- Tasks
- Event Correlation Management
- Correlation Composer
- Read the information in the *Disabling a Correlator* topic.

2. Access the Correlator Composer main window and disable the following three correlators. You must disable *all* of them:

- OV\_NodeIf\_NodeDown
- OV\_NodeIf\_NodeNotConnector
- OV\_NodeIf\_PrimaryIFUnknown

## Correlator Fact Store Files

Correlation Composer supports the use of multiple factstore files. Each factstore file contains a logical grouping of correlators. To access the contents of a factstore file from Correlation Composer, the factstore file

must be associated with a namespace in the `namespace.conf` configuration file. The factstore files and the `namespace.conf` file are located in:

*Windows:* `\NNM_install_dir\ecs\CIB\namespace.conf`

*UNIX:* `$OV_CONF/ecs/CIB/namespace.conf`

For more information about Correlation Composer configuration files, see *HP OpenView Correlation Composer's Guide*.

If you are planning to make experimental changes to the correlator parameter settings, you may wish to make a backup of the factstore files before proceeding.

When correlator configurations are deployed, all correlator factstore files are merged within one NNM factstore file:

*Windows:* `\NNM_install_dir\conf\ecs\circuits\Composer.fs`

*UNIX:* `$OV_CONF/ecs/circuits/Composer.fs`

You should never edit this file directly; only make changes through the Correlator Composer window.

All factstore files, except `Composer.fs`, are automatically included when `ovbackup` and `ovrestore` are run (see “Backup/Restore to Protect Your Investment of Time” on page 149).

## Troubleshooting

For troubleshooting information about the Correlation Composer, see the following references:

- Access the following pdf format manual from the NNM main window, select `Help:Documentation`:
  - *HP OpenView Correlation Composer's Guide*
- Access the following pdf-format white paper file: `Developing_NNM_Event_Reduction.pdf`
  - *Windows:* `NNM_install_dir\Doc\WhitePapers\`
  - *UNIX:* `$OV_DOC/WhitePapers/`

This information explains how to ensure that you do not break existing event reduction implementations. It also explains the process required for carefully merging your new correlators into the NNM correlator fact store file.

You *might* need to install the white papers on Windows-based NNM management stations. See “White Papers” on page 57.

- The `TroubleshootingEventReduction.txt` file explains how to create event logs and play them back into the NNM event correlation system for testing purposes:

— *Windows:* `NNM_install_dir\contrib\ecs\`

— *UNIX:* `$OV_CONTRIB/ecs/`

You *might* need to install the contrib files on Windows-based NNM management stations. See “The contrib Directory” on page 56.

## Creating Additional Correlators

---

### NOTE

All de-duplication configurations, correlators defined within Correlation Composer, and all correlations defined within ECS must *work well together* on your NNM management station. Before you begin, it is critical that you print out and read the following white paper:

`Developing_NNM_Event_Reduction.pdf`

- *Windows:* `NNM_install_dir\Doc\WhitePapers\`
- *UNIX:* `$OV_DOC/WhitePapers/`

This information explains how to ensure that you do not break existing implementations.

You *might* need to install the white papers on Windows-based NNM management stations. See “White Papers” on page 57.

---

You must use Correlation Composer in developer mode in order to create new correlators.

Correlation Composer provides templates for the most commonly used correlation logic. These templates make it easy for you to develop your own:

- **Enhanced**  
Used to trigger the creation of one or more new alarms or to augment the information content of an alarm.
- **MultiSource**  
Used to define a relationship between an arbitrary number of alarms, potentially from different sources, that together form a logical set that identifies a problem. The set of alarms must all arrive within a defined time period. Upon completion of the set, the alarms can be discarded, modified, or a new more meaningful alarm can be generated.
- **Rate**  
Used to measure the number of incoming events within a defined time period. When a specified number is received, you can choose to discard the alarms and generate a new more meaningful alarm.
- **Repeated**  
Used to either discard duplicate alarms within a defined time period, or generate a new alarm each time an additional alarm is received so that the current number of alarms received can be specified in the alarm message text.
- **Suppress**  
Used when a specific category of alarms needs to be discarded, so that they never appear in the NNM Alarm Browser.
- **Transient**  
Used to detect a defined number of paired event occurrences within a defined time period, such as node up/node down. The paired events can be discarded, and a new more meaningful alarm can be generated.

---

**NOTE**

Support for the correlator template named **User Defined** is not included. If you wish to use the Correlation Composer **User Defined** template, contact HP to purchase a Partner Care Extended support contract before beginning.

---

Correlators can be chained together to accomplish complex tasks (see “Router/Switch Health” on page 377 for an example of correlator groups).

If you wish to use external calls, contact HP to purchase a Partner Care Extended support contract before beginning.

Print out and read the following reference materials:

- Access the following pdf format manual from the NNM main window, select `Help:Documentation`:

— *HP OpenView Correlation Composer's Guide*

- Find the *csmerge* reference page in NNM's online help (or the UNIX manpage) for information about Correlation Composer's tool for merging new correlators (that are developed and tested in an isolated environment) back into the NNM correlator fact store file on your production NNM management station.

- Access the following pdf-format white paper file:

`Developing_NNM_Event_Reduction.pdf`

— *Windows:* `NNM_install_dir\Doc\WhitePapers\`

— *UNIX:* `$OV_DOC/WhitePapers/`

This information explains how to ensure that you do not break existing event reduction implementations. It also explains the process required for carefully merging your new correlators into the NNM correlator fact store file.

- The `TroubleshootingEventReduction.txt` file explains how to create event logs and play them back into the NNM event correlation system for testing purposes:

— *Windows:* `NNM_install_dir\contrib\ecs\`

— *UNIX:* `$OV_CONTRIB/ecs/`

Developing your own event correlators is a powerful option, but a non-trivial task. You should consider carefully whether this is something you want to do or whether it might be quicker, simpler, and cheaper to contract with a specialist to supply the correlator or correlators you need. Contact HP Consulting Service (see “HP Consulting Service” on page 58) or other system integrators for assistance.

To develop a correlator you need to:

- Learn to develop correlators through Correlation Composer training and/or documentation.
- Design and debug the correlator on an isolated test machine.



- If you are using the NNM Advanced Edition, deploy the correlator to each NNM management station and collection station, as described in the `Developing_NNM_Event_Reduction.pdf` white paper.

### When to Design Your Own Event Correlators

There are several key factors which indicate that a new correlator is required, or that an existing correlator needs to be reconstructed:

- No existing correlator provides the correlation you need.
- Parameters of an existing correlator need to be set to values that are outside the minimum and maximum allowed.
- The correlator requires access to external data (for example, to look up an inventory database) and therefore needs to use ECS annotation.
- Special event I/O requirements exist.

### What is Involved

To design an effective correlator, you must:

- Understand typical network event patterns and what they mean.
- Know the architecture of your network management system.
- Identify external data access (ECS Annotation) requirements.
- Define the problem concisely in terms of filtering and time-related if-then statements.
- Determine if all data is available from the event's variable-bindings. If not determine if:
  - The external data is subject to change or is relatively stable.
  - The required data is small enough to store in memory.
  - The access speed and reliability is sufficient when extracting from your database (particularly if the database is remote).

Then, to develop the correlators, you need to:

- Use Correlation Composer in developer mode.
- Set up a test platform.

## Correlation Composer Correlators

- Develop sample logs of events. See the following file for information about how to create log files of events, and how to feed them into the NNM event correlation system in your test environment:
  - *Windows:*

```
NNM_install_dir\contrib\ecs\  
TroubleshootingEventReduction.txt
```
  - *UNIX:*

```
$OV_CONTRIB/ecs/TroubleshootingEventReduction.txt
```
- Your NNM support contract includes coverage for the use of the following Correlation Composer templates (provided that you do not invoke calls to APIs, libraries, scripts, or programs that are not included as part of the Correlation Composer product embedded within NNM):
  - Enhanced
  - MultiSource
  - Rate
  - Repeated
  - Suppress
  - Transient

If you wish to use external calls, contact HP to purchase a Partner Care Extended support contract before beginning.

---

### NOTE

Support for the correlator template named **User Defined** is not included. If you wish to use the Correlation Composer **User Defined** template, contact HP to purchase a Partner Care Extended support contract before beginning.

---

## What is an SNMP Variable-Binding and How Do I Identify One?

When an SNMP event or trap is generated, a group of *variables* are included in the data. These variables are defined within the MIB file from which the event or trap originates. These variables are often referred to as *var-binds* because they are bound to a particular instance of the SNMP event or trap. The *SNMP variable-binding list* associated with a particular trap or event can provide a wealth of valuable information for you to use when creating your own event reduction strategies using NNM de-duplication, ECS correlation, or Correlation Composer correlator. This section explains a couple of ways to determine the available SNMP variable-bindings:

- “Look within the MIB File” on page 387.
- “Check the Alarm Message Text in the Alarm Browser” on page 388.

This section also explains the SNMP variable-bindings contained within the NNM *interface-status* and *node-status* alarms on page 388, because they can be very useful when defining event reduction implementations.

### Look within the MIB File

The most direct way to determine the available SNMP variable-bindings is to locate the MIB file, open the MIB file with an ASCII editor, and look up the variable list for that particular trap or event. Check the documentation that came with your router or switch to locate the MIBs currently in use. If you do not have a copy of the MIB file, try using the MIB Depot web site. It may be a good resource for obtaining MIBs:

<http://www.mibdepot.com>

Otherwise, contact the vendor who created that particular MIB to obtain a copy.

## Check the Alarm Message Text in the Alarm Browser

The alarm messages that display within the NNM Alarm Browser are controlled by the NNM Event Configuration settings. Usually the alarm message text includes a specified subset of available SNMP variable-binding values, as defined through the NNM Event Configuration window.

---

### NOTE

For more information about NNM's Event Configuration feature and the choices for controlling alarm message text, see "Event Configuration Overview" on page 404. The event configuration settings are stored in the `trapd.conf` file.

---

You can easily modify the alarm message configuration settings. See "Variables and Special Characters Allowed" on page 418 for information about alarm message configuration choices. Some very useful alarm message variables are:

`$$` = prints out the total number of SNMP variable-bindings defined within the originating MIB of the incoming trap

`$$*` = prints out the value passed with each of the defined SNMP variable-bindings

You can use the alarm message information as a starting point for your event reduction strategy, even if you do not have a copy of the MIB.

Any incoming traps whose MIBs have not been loaded into NNM (see "Loading MIBs in the MIB Database" on page 395) are displayed as generic traps called `EnterpriseDefault (.1.3.6.1.4.1.*)`. The alarm message for the "generic" trap is defined as follows:

```
no format in trapd.conf. $$ args: $$*
```

## Special SNMP Variable-Bindings within NNM Interface and Node Status Alarms

The OpenView interface and node status events include a wealth of information transported in SNMP variable-bindings. They are of particular interest when developing new de-duplication configurations, correlations, and correlators.

---

**NOTE**

The first variable in the SNMP variable-binding list is traditionally considered to be var-bind “1” (one). If you are developing a Correlation Composer correlator, the first variable in the SNMP variable-binding list is var-bind “0” (zero).

---

### **OpenView Interface Status Variable Bindings**

- **Var-Bind 1:** The ID of application sending the event.
- **Var-Bind 2:** The hostname of the node that caused the event.
- **Var-Bind 3:** The HP OpenView object identifier of the node that caused the event.
- **Var-Bind 4:** The database name, if applicable.
- **Var-Bind 5:** A time stamp for when the event occurred.
- **Var-Bind 6:** The HP OpenView object identifier of the interface that caused the event.
- **Var-Bind 7:** The name or label for the interface that caused the event.
- **Var-Bind 8:** The IP address for the interface that caused the event, or "0" if unavailable.
- **Var-Bind 9:** The NNM management station ID.
- **Var-Bind 10:** Event correlation event UUID which identifies the primary failure.
- **Var-Bind 11:** Number of bits in the subnet mask of interface.
- **Var-Bind 12:** Locale neutral description of the interface (typically `ifAlias`). Note that the regular label for the interface is transmitted in var-bind 7 and so the presence of this new variable-binding allows the interface name (for example, `lan0`) and the interface `ifAlias` to be displayed simultaneously.
- **Var-Bind 13:** Locale neutral comma separated list of capabilities of the interface. For example, `isSwitch,isIPRouter`. Not all capabilities are included that may be available in NNM. The primary purpose of this interface is to identify connector devices within event reduction implementations.

## What is an SNMP Variable-Binding and How Do I Identify One?

- **Var-Bind 14:** Name of the primary failure host if available.
- **Var-Bind 15:** Name of the primary failure entity if available.
- **Var-Bind 16:** The HP OpenView object identifier of primary failure entity if available.
- **Var-Bind 17:** Locale neutral description of the primary failure entity if available.
- **Var-Bind 18:** Locale neutral comma separated list of capabilities of the primary failure entity if available.

Variable-binding 14 through 18 describe the primary failure. For example, node C may be inaccessible due to a failure on router R. Node R would be the primary failure and node C would be the secondary failure. If node R is a primary failure then variable-binding 12 through 15 will describe node R and all the other variable-binding describe node C. If node C is a primary failure then all the SNMP variable-bindings describe node C. See “Connector Down Correlation” on page 350 and “Secondary Failure Polling” on page 177 for more information about designating primary and secondary failures.

### OpenView Node Status Variable Bindings

- **Var-Bind 8:** Locale neutral comma separated list of capabilities of the entity. For example, `isSwitch, isIPRouter`.
- **Var-Bind 9:** Name of the primary failure host if available.
- **Var-Bind 10:** Name of the primary failure entity if available.
- **Var-Bind 11:** The HP OpenView object identifier of primary failure entity if available.
- **Var-Bind 12:** Locale neutral description of the primary failure entity if available.
- **Var-Bind 13:** Locale neutral comma separated list of capabilities of the primary failure entity if available.

---

---

# 12

# Customizing Events: Doing It Your Way

NNM allows you to take control of the event system through the use of several built-in applications. This chapter explains the purpose and use of each of these standard NNM applications:

- *SNMP MIB Browser* (page 394) and *Load/Unload MIBs* (page 395). Allows you to load and explore the functionality of the MIBs of your choosing. Once the MIB is loaded, you can query to determine the remote device's status, configuration, and available resources. You can also *set* MIB object values, provided you configured your network's community names into NNM.
- *DMI Browser* (page 399) and mapping MIFs (page 401). *NNM for the Windows operating system only*: Allows you to explore and access the functionality of a network device's Desktop Management Interface (DMI) service provider to which NNM subscribes. You can query to determine the remote device's status, configuration, and available resources.
- *Event Configuration* (page 404). Allows you to customize events, such as changing standard wording of MIB-generated error messages to more meaningful messages for your organization. You can control which Alarm Browser category the event is assigned to and also configure:
  - Automatic Actions, such as dialing a pager, that execute whenever NNM receives a specific event. An Automatic Action command is tied to a particular event and can be limited to a specified list of IP hostnames and/or IPX addresses.
  - Additional Actions that provide access to batch files or executables that your team uses occasionally. You configure the additional actions to show up in NNM's menu bar. Additional Actions can be applied to any alarm or specified groups of alarms.
- Information gathering applications:
  - *MIB Application Builder* (page 425). Allows you to set up new menu items within NNM that provide access to your choice of useful MIB objects on an as-needed basis.



- *Data Collection & Thresholds* (page 429).  
Allows you to configure NNM to gather your choice of SNMP-MIB objects automatically (scheduling specific collections, establishing threshold monitoring of devices, and controlling what shows up in the Alarm Browser as a result).
- *MIB Expressions* (page 442).  
You can create MIB expressions (mathematical formulas using MIB objects for analysis) to be used by Data Collection & Thresholds.

## SNMP MIB Browser

The SNMP MIB Browser is available from any submap under `Tools:SNMP MIB Browser`. The SNMP MIB Browser is useful for:

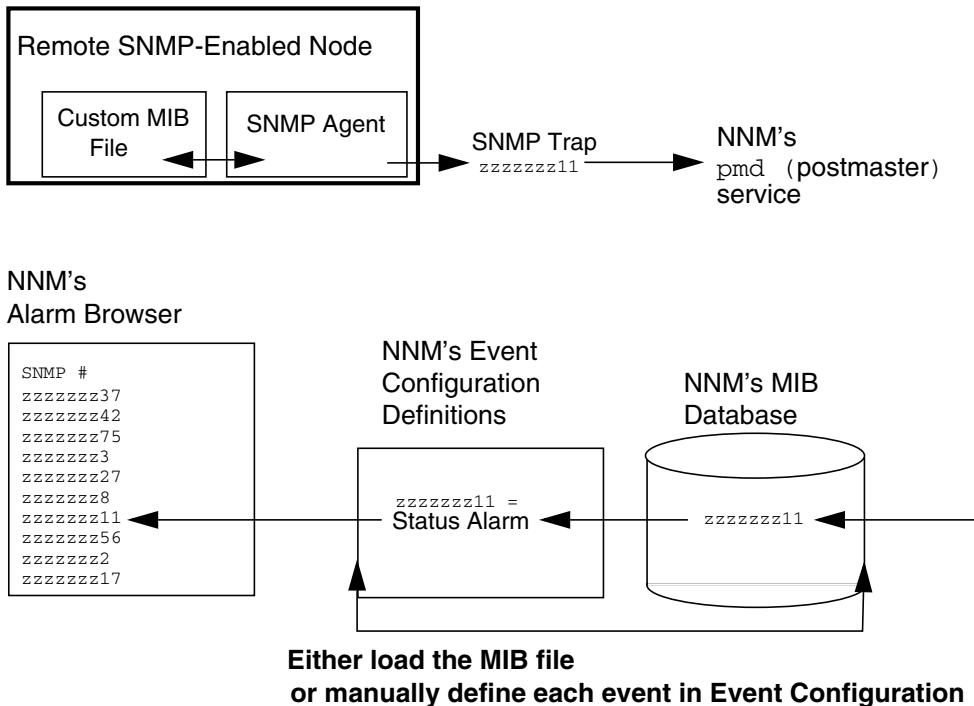
- Exploring your MIBs and their objects to become familiar with the network management capabilities that they provide. It serves as a useful reference tool. Type in any MIB number and view the description of what it is intended to do. Click on any leaf in any loaded MIB tree to view the MIB object's description.
- Querying MIB objects and viewing the returned instances for a network device that you selected on the submap or specified by typing in the IP address. You can view real-time MIB values for a selected node. You can also collect multiple instances by specifying a time interval for queries and display a graph of the results. The node for which you want to view MIB objects must support SNMPv1 or SNMPv2C.
- Setting MIB values on a network device. Prerequisites are:
  - The selected node on which you want to set MIB objects must support SNMPv1 or SNMPv2C.
  - The SNMP agent on the selected node must be configured to respond to SNMP SetRequests.
  - You must know the agent's SET-community name.

For more information about what you can do with NNM's SNMP MIB Browser, see the online help available within the SNMP MIB Browser window.

## Loading MIBs in the MIB Database

When you install NNM, the standard SNMP version 1 and 2 MIBs are automatically loaded and configured for your convenience. Often devices on your network have their own custom vendor-specific MIBs that allow for monitoring specific aspects of that vendor's devices, and allow you to customize NNM to proactively manage the devices. If you are running extensible SNMP agents on your devices, the agents may be using custom MIB files. Before you can take advantage of these custom MIB capabilities, the MIB definition loaded into NNM's MIB database must match the version of the MIB in use on the device itself.

**Figure 12-1** Configuring Custom MIB File Definitions



---

**NOTE**

To obtain the most recent version of a MIB file, check NNM's installation CD, contact the vendor who wrote the MIB file, or search the WWW (<ftp://ftp.isi.edu/mib/>) for quick access to the MIB file. Make sure the MIB definition loaded into NNM's MIB database matches the version of the MIB in use on the device itself.

---

The NNM installation CD includes hundreds of vendor-specific MIBs for your convenience. Simply load the MIBs that are relevant and useful for your specific network configuration. When you load a MIB into NNM's MIB database, that MIB's trap definitions are automatically added to the Event Configuration application so that you can customize NNM's use of each one. Use NNM's Event Configuration to identify which traps are forwarded to NNM's Alarm Browser. (See the "Event Configuration Overview" on page 404 for more information.)

Once a MIB is loaded, you can use NNM's Data Collection & Thresholds application to configure threshold monitoring and set scheduled collection intervals for the MIB objects of your choice.

## Prerequisites

Before loading a MIB for a device, ensure that these prerequisites are met:

- The new device must be installed on the network according to the installation instructions provided by the vendor. The SNMP agent running on the device must be configured to forward traps to the NNM management station.
- You must obtain a copy of the current MIB definition file from the vendor or NNM's installation CD.
- Locate the MIB file (on NNM's installation CD, all MIB definitions are stored in the `snmp_mibs` subdirectory). To keep the MIB files sorted logically, the best practice is to create a subdirectory on the management station labeled with the vendor's name. Then copy the MIB file to that subdirectory on the management station before loading it. For example:

— *Windows:* `install_dir\snmp_mibs\vendor\MIB`

— *UNIX:* `$OV_SNMP_MIBS/vendor/MIB`

- The MIB must conform to *RFC 1155*, *RFC 1212*, *RFC 1902*, *RFC 1903*, or *RFC 1904*.
- If you have unloaded all the MIBs from the management station, it is recommended that you load MIB-II before you load an enterprise-specific MIB.

## Procedure Tips

---

### TIP

Close NNM's Event Configuration window prior to loading any MIB file. NNM automatically establishes a preliminary event configuration for you of each trap in the new MIB (provided the Event Configuration window is closed).

---

To load a MIB:

1. Place the NNM installation CD into the management station's CD drive. By default, NNM assumes you want a MIB from the installation CD. If the enterprise-specific MIB that you want to load is located elsewhere, you can enter the full path name to the file after opening the Load/Unload MIB window (in the field labeled MIB File to Load).
2. From any submap, select `Options:Load/Unload MIB:SNMP`. See NNM's online help from within the Load/Unload MIB:SNMP window for more information.
3. If the MIB file contains trap definitions, a dialog box appears asking if you would like to establish generic configurations for each trap in NNM's Event Configuration.

You can use the `Tools:SNMP MIB Browser` operation to view the new MIB objects. See "SNMP MIB Browser" on page 394.

If a MIB does not load properly, you will receive an error message. The problem could be:

- A syntax error in the MIB.  
The Load/Unload MIB program's error message identifies the MIB object where the error is occurring.

- Mismatched versions of the vendor's MIB.

For example, the vendor's MIB on the NNM installation CD might not match the installed MIB version on the vendor's device on your network. In this case, obtain the MIB version that matches the MIB currently on the device.

In either case, contact the vendor to resolve the problem.

---

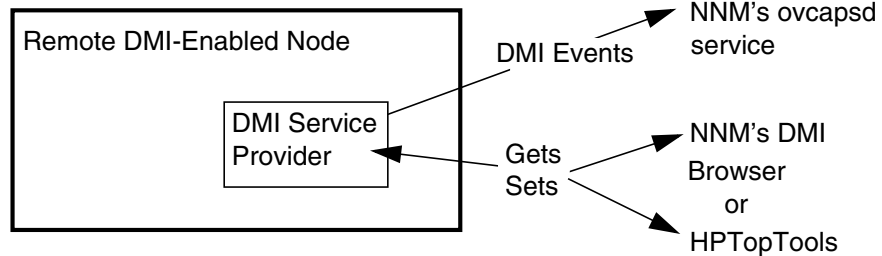
## DMI Browser

The DMI Browser is available on NNM management stations running a Windows operating system. From any submap under `Tools:DMI Browser`. The DMI Browser allows you to view real-time MIF values for the DMI-enabled device currently selected on your submap.

### Creating DMI Queries

In addition to DMI events, DMI also supplies GET and SET capabilities for some attributes.

**Figure 12-2**      **Queries to a DMI Service Provider**



- *DMI version 2:*  
To run DMI queries of devices running any DMI 2.0 service provider, use NNM's DMI Browser application. See the online help from within the DMI Browser for more information.
- *DMI version 1.1:*  
To run DMI queries of devices running the HP-RDMI service provider, use the HPTopTools application provided in the contributor's subdirectory on the NNM installation CD. See the readme file in the contributor's directory (see page 56).

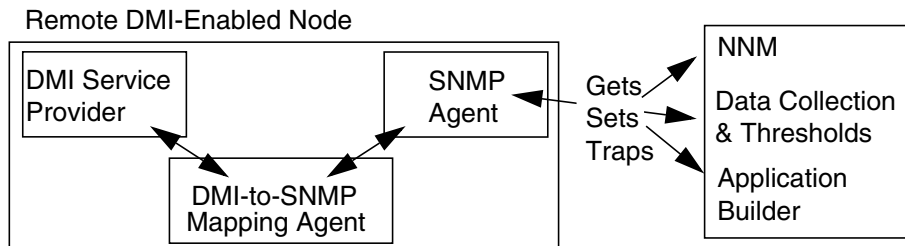
---

**NOTE**      As an alternative, rather than communicating directly with the DMI service provider on your device, you can communicate with the DMI service provider indirectly through an SNMP agent installed on the

device along with a DMI-to-SNMP mapping agent. (Intel provides a mapping agent in their DMI 2.0 Service Provider Software Development Kit.) See the following figure.

The SNMP agent then handles all communications. Therefore, NNM tools such as the MIB Browser, MIB Application Builder, Data Collection & Thresholds application, and SNMP Grapher can be used to query DMI data via the DMI-to-SNMP mapping agent. NNM no longer subscribes to the DMI service provider to receive DMI events from devices running a DMI-to-SNMP mapping agent.

**Figure 12-3** Bypassing a DMI Service Provider with an SNMP Agent

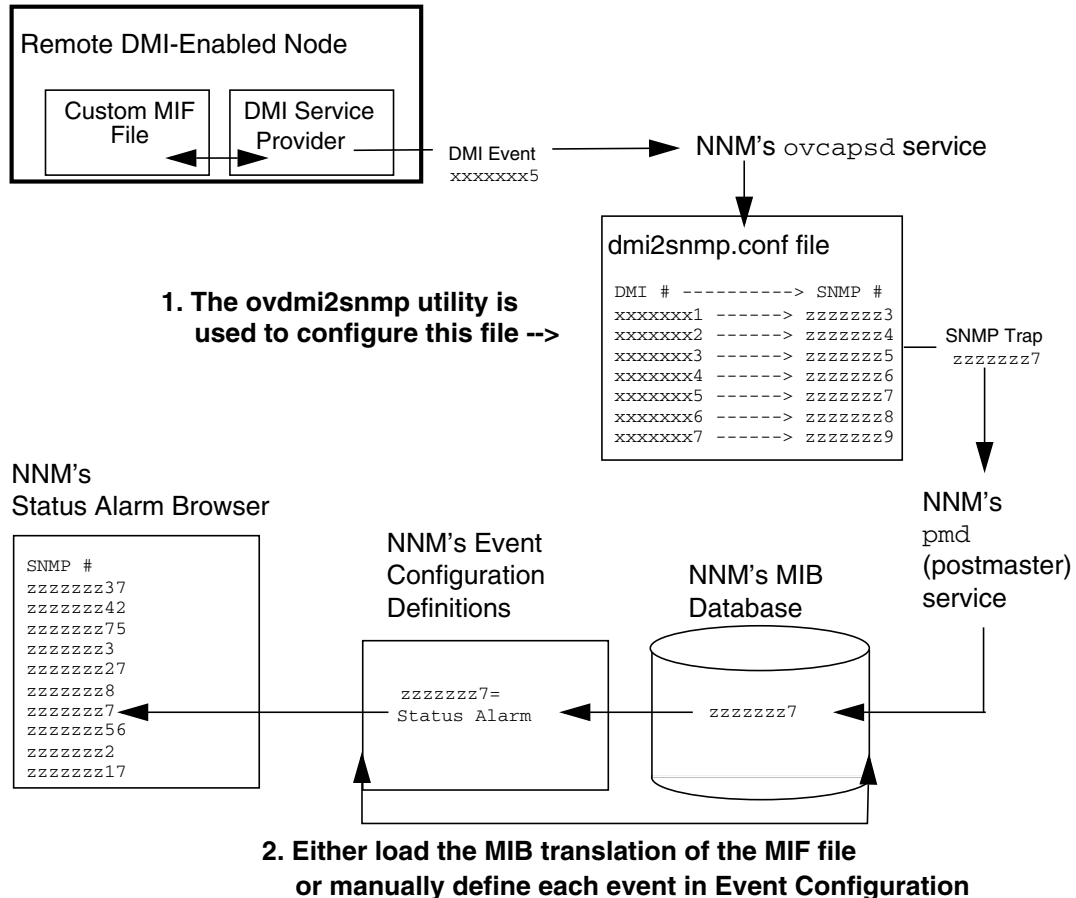




## Loading DMI-to-SNMP Event Mappings

If your DMI service provider software is using a vendor-specific MIF instead of or in addition to the DMTF's standard MIF, you must configure NNM to understand the custom MIF events. Otherwise, a generic DMI event message (based upon the `dmiEventIndication` definition) is posted to the Alarm Browser each time the vendor-specific DMI event is received. To customize NNM's treatment of these events, you need to complete several steps, as shown in the following figure.

**Figure 12-4** Configuring Custom MIF File Definitions



## If a MIB Translation of the MIF File Is Available

The vendor of the DMI MIF may provide an SNMP MIB translation for your use. This allows you to load the MIB file and work with the event configurations in the same manner as you work with any SNMP MIB. However, an additional step is required. You must also provide the `ovcapsd` service with the new mapping information so that it can convert the incoming DMI event message to the correct SNMP trap message.

1. Update `ovcapsd`'s mapping file.

A tool is provided to assist with this task. See the `ovdmi2snmp` reference page in NNM's online help for information about how to do this step. You will need to know each of the MIF's Event Generation Classes and their:

- Associated class data.
- Corresponding SNMP trap enterprise object identifier.

2. Load the vendor-provided MIB file into NNM.

From any submap, select `Options:Load/Unload MIB`. Best practice is to place the MIB file into the following directory, which you create:

- *Windows:* `install_dir\snmp_mibs\DMIVendors\*.*`
- *UNIX:* `/var/opt/OV/share/snmp_mibs/DMIVendors/*`

When prompted to load trap definitions, select `YES` to automatically establish the default configuration for each DMI event in NNM's Event Configuration.

3. Use NNM's Event Configuration to review and customize each new trap. See "Event Configuration Overview" on page 404.

## If No MIB Translation of the MIF File Is Available

If no SNMP MIB translation of your MIF file is available for your use, you can manually set up NNM's `ovcapsd` background process and NNM's Event Configuration with the new mapping information so that it can convert the incoming DMI message to the correct SNMP trap message.

---

**TIP**

---

MIF-to-MIB translation tools that convert any MIF file to a MIB file are available. Intel provides one in their DMI 2.0 Service Provider Software Development Kit.

1. Update *ovcapsd*'s mapping file.

A tool is provided to assist with this task. See the *ovdmi2snmp* reference page in NNM's online help for information about how to do this step. You will need to know each of the MIF's Event Generation Classes and their:

- Associated class data.
- Unique SNMP trap enterprise object identifier for the DMI Event Generation Class, which you choose.
- SNMP trap ID, which is equal to the DMI Event ID.

If your company has a registered "enterprise branch" assigned through the Internet Assigned Numbers Authority (IANA), use one of those available numbers. Otherwise, you may create a new OID that is unique to your installation under the `hpOVUserDefnEvents` branch (1.3.6.1.4.11.2.17.9.xxxxxx).

2. Use NNM's Event Configuration to create a new trap configuration manually for each DMI event in your MIF file. Be sure to use the same corresponding SNMP trap enterprise object identifier and event/trap ID that you used in step 1 for each event. See "Event Configuration Overview" on page 404.

## Event Configuration Overview

SNMP events are the building blocks of network management. Alarms are generated when NNM receives certain events. Thoughtful planning and use of event configuration will allow NNM to monitor your network effectively, taking the pressure off of you and your team. The Event Configuration application enables you to control and enhance the way in which NNM handles events. This section describes the following few suggestions among numerous possibilities:

- Control alarm message posting and text.
  - Control which alarms are posted into NNM’s Alarm Browser and which alarms are ignored.
  - Assign an Alarm Browser category for the event’s alarm.
  - Customize alarm messages to make them more meaningful or to provide specific troubleshooting information to your team.
- Establish automatic actions for NNM to perform whenever a specific event is received (such as dialing a pager).
- Place access to a commonly used batch file command or any executable (such as launching commonly used troubleshooting tools) on NNM menus for your team’s convenience.

For more ideas about customizing NNM’s event system, review the choices in the Event Configuration window and refer to NNM’s online help available from within the Event Configuration window.

### Prerequisites

- The MIB for which you want to configure events must be loaded into NNM’s MIB database. From any submap, select `Options:Load/Unload MIBs:SNMP` to display the list of all MIBs currently in NNM’s MIB database. If the MIB you wish to use is not loaded, see “Loading MIBs in the MIB Database” on page 395.
  - A generic configuration for each trap from the MIB is automatically set up in NNM’s event configuration feature when you load the MIB.

- MIB objects that are not *traps* can be manually set up in NNM's Event Configuration program. You can then generate GET, SET, or data collection events based upon the new event configuration.
- Make sure that you understand the enterprise-specific MIB. To configure events, you need to understand the trap definitions for the device and what they do. Most device vendors include documentation about their enterprise-specific traps with their product. This documentation typically describes the trap and when the trap is generated. In addition, some vendors include overview information describing strategies for how to manage their specific device. The vendor documentation can give you the conceptual understanding that you need to configure effective event formats and actions.

## Event Configuration Window

---

### TIP

You cannot configure events from the NNM's web interface. However, the web-based Alarm Browser displays the alarms resulting from your event configurations.

---

To display the Event Configuration window:

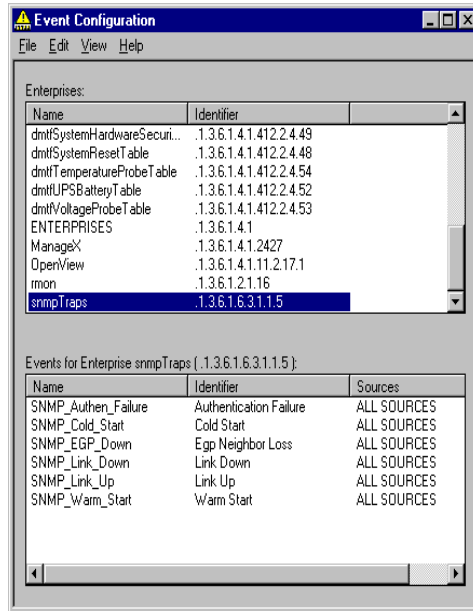
- From any Alarm Browser window, select `Actions:Configure Event`.
- From any submap, select `Options:Event Configuration`.

The Event Configuration window contains an Enterprise Identification section, an Event Identification section, and a menu bar.

To close the Event Configuration window, select `File:Close`. Be sure to save your changes as you exit. Event configurations are stored in the `trapd.conf` file.

Figure 12-5 shows the Event Configuration window.

**Figure 12-5** Event Configuration Window



### Enterprise Identification List

The MIB files that define specific events are supplied by a number of vendors (enterprises). The list in the top half of the Event Configuration window identifies the enterprise that supplied the MIB whose event you wish to configure. For example, if you are configuring NNM to communicate with a Cisco Router's SNMP agent, then the event configuration should be defined under the corresponding Cisco enterprise identification (.1.3.6.1.4.1.9.).

Events generated by NNM processes (such as netmon or snmpCollect) will use the HP OpenView enterprise identifier.

---

#### NOTE

If you are configuring a threshold event created by the HP OpenView SNMP Data Collector, you should configure it under the OpenView enterprise name (.1.3.6.1.4.1.11.2.17.1), regardless of which node the Data Collector is polling. Even if the event is for a Cisco device

exceeding a threshold, the enterprise ID of the event will be `OpenView`, not `Cisco`, since the event is coming from the HP OpenView SNMP Data Collector.

---

Select the enterprise associated with the events you want to configure. Each item in the list contains:

Enterprise Name

A convenient and meaningful representation of the Enterprise ID used by the Event Configuration application. This label usually corresponds to the enterprise name as defined by the enterprise-specific MIB.

Enterprise ID

The system object ID in dotted notation. The ID corresponds to the value supplied with the trap. If no traps are defined for an enterprise, the application will generate a generic message using the `ENTERPRISES` enterprise. `ENTERPRISES` acts as a default.

You can add or delete enterprises using:

- *Windows*: The `Edit:Events` menu item.
- *UNIX*: The `Edit:Configure` menu item.

### Event Identification

The list in the bottom half of the Event Configuration window identifies events associated with the enterprise selected in the top half. By default, NNM configures all standard SNMP trap definitions and automatically supplies a generic configuration of all traps present in any MIB that you load into the MIB database. You may configure any event (trap, GET, SET, or custom data collection event). If no enterprise is selected, the list will be blank.

Event Name

The name used to reference the event (can be anything meaningful to you).

## Event Identifier

The event identifiers may be shown as generic and specific traps or as object identifiers, depending on how you have the `View:Event Identifiers` menu set up.

If the menu is set up to display the events as SNMP traps, then the generic traps 0 through 5 are defined by SNMP and assume a specific trap of 0. The generic trap 6 is the enterprise-specific trap. `EnterpriseDefault` provides default trap handling for all traps for an enterprise. The specific traps are defined for this enterprise. The vendor defines what the specific trap numbers mean.

The object identifier for an event is in dotted decimal format. For enterprise-specific events, this identifier is the concatenation of an enterprise object ID, 0, and the specific number (for example, `1.3.6.1.4.1.11.2.17.1.0.58916872`). For SNMP generic traps (0-5), this identifier is the concatenation of the `snmpTraps` enterprise object ID and the generic trap number incremented by one (for example, `1.3.6.1.6.3.1.1.5.5` for the `SNMP_Authen_Failure` generic trap).

The default is to view the events as object identifiers.

## Sources

The potential sources for the event (such as nodes), or `ALL SOURCES` if this event is to be used regardless of the source of the event.

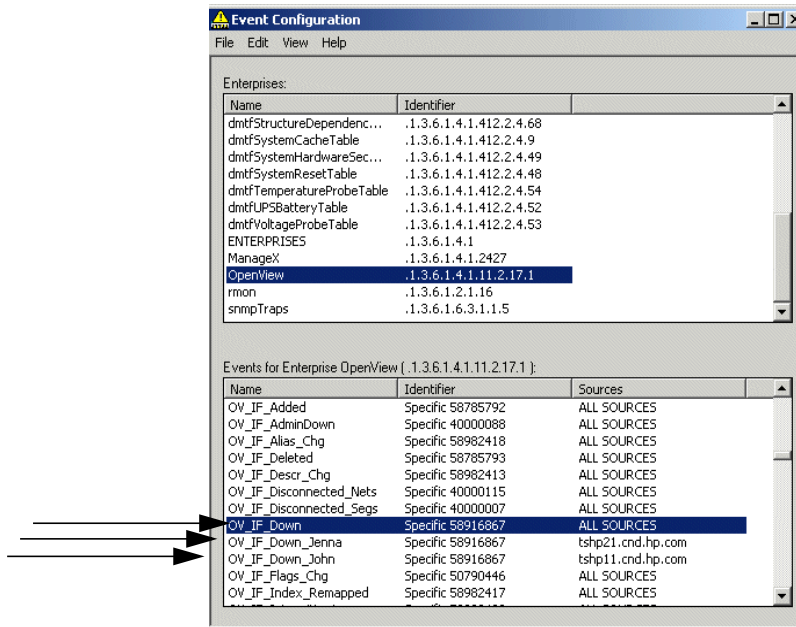
NNM allows you to configure events in a number of ways and even add new events. The preconfigured event definitions provide a starting point for your use.

For example, you can use the `OV_IF_Down` MIB object to monitor your most critical routers. For this example, assume that John is in charge of the routers at site A, and Jenna is in charge of the routers at site B. You can copy the preconfigured `OV_IF_Down` and rename it to `OV_IF_Down_John`; change the source field to indicate router interfaces for which John is responsible; then configure NNM to page John whenever a router for which he is responsible goes down. Now repeat the procedure and create another event configuration that you name



OV\_IF\_Down\_Jenna; change the source field to indicate router interfaces for which Jenna is responsible; then configure NNM to page Jenna whenever a router for which she is responsible goes down.

**Figure 12-6** Copying Preconfigured Events to Create New Ones



Vendors provide MIB objects that allow you to take this concept a step further and proactively monitor performance on your critical devices. You can configure events to alert you to potential trouble, rather than waiting for a device to fail and send a trap to the management station.

You can add, modify, copy, or delete events using:

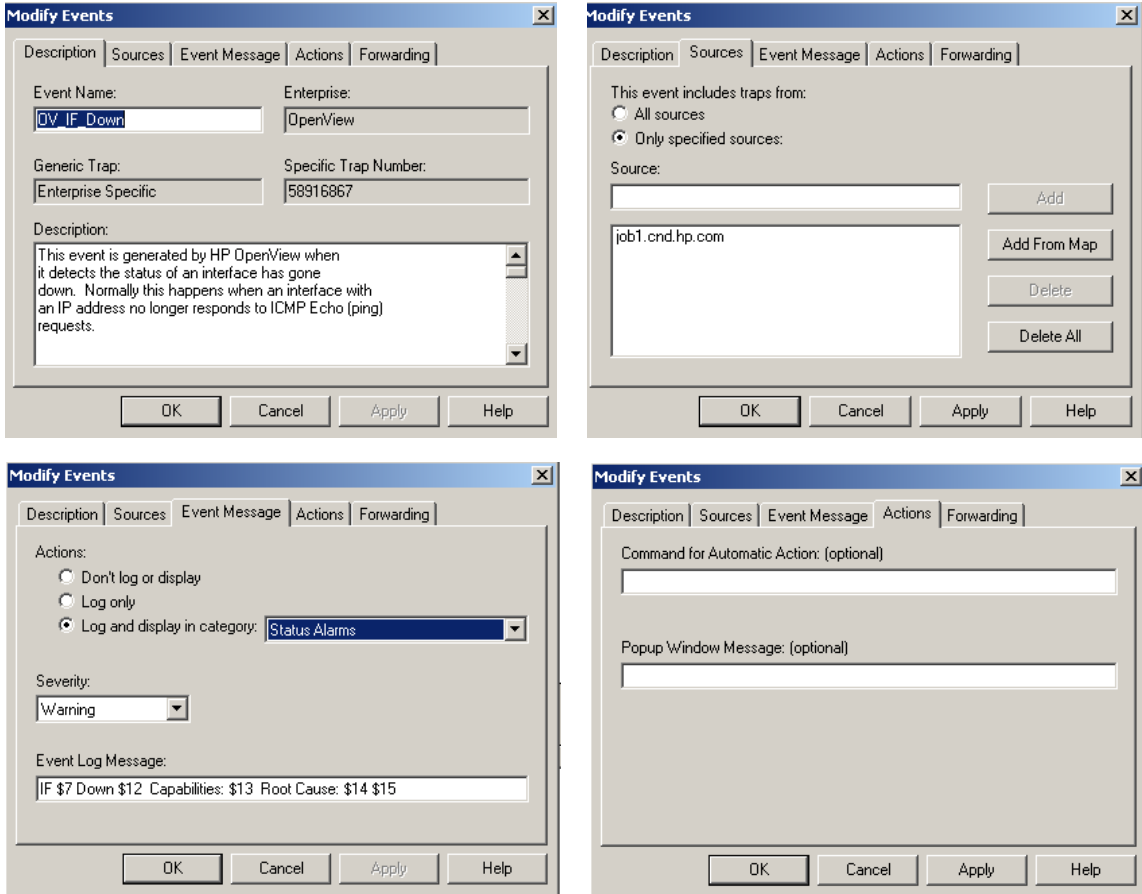
- *Windows*: Select the event, then select the `Edit:Events` menu item.
- *UNIX*: Select the event, then use the `Edit` menu.

Events can be customized in a wide variety of ways to meet your team's needs. Review the choices in the Event Configuration window and refer to NNM's online help available from within this window for more information and ideas.

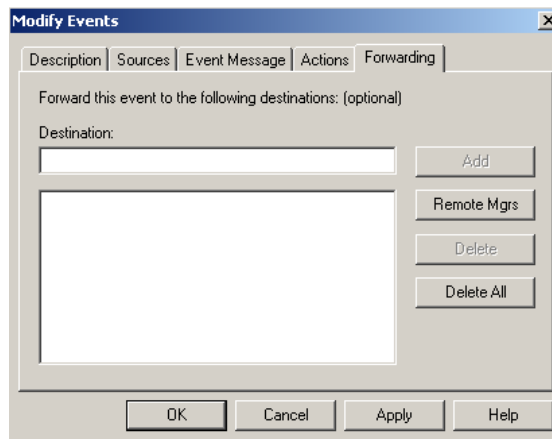
The customizations that you make to event configurations are written to the `trapd.conf` file. Refer to the *trapd.conf* reference page in NNM's online help (or the UNIX manpage) for more information.

This section covers *only a few* of the possibilities for configuring events.

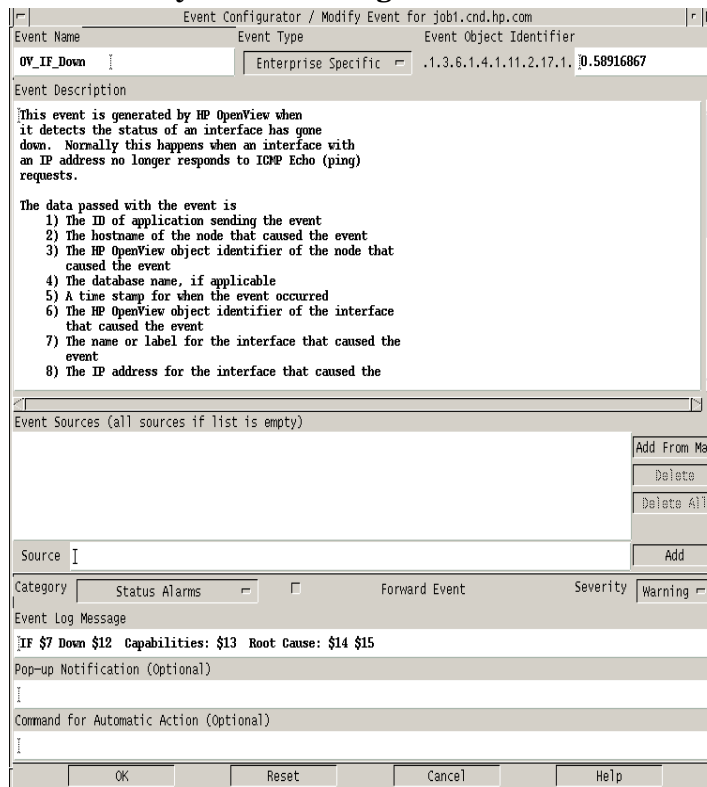
**Figure 12-7** Windows: Modify Events Property Sheet (5 tabs)



**Figure 12-8 Windows: Modify Events Property Sheet, continued**



**Figure 12-9 UNIX: Modify Events Dialog Box**



## Controlling Alarm Message Posting and Text

You can control how NNM handles specific alarms, such as:

- Control which alarms are posted into NNM's Alarm Browser and which alarms are ignored.
- Assign an Alarm Browser category for the event's alarm.
- Customize alarm messages to make them more meaningful or to provide specific troubleshooting information to your team.

### Selecting Events to Post in the Alarm Browser

1. Ensure that the prerequisites have been met (see "Prerequisites" on page 404). Then, open the Event Configuration window:
  - From any Alarm Browser window, select `Actions:Configure Event`.
  - From any submap, select `Options:Event Configuration`.
2. In the top half of the window, select the enterprise that supplied the MIB file that defines the SNMP event that you wish to configure. In the bottom half of the window, select the specific SNMP event:
  - *Windows*: Select `Edit:Events->Modify`, then select the Event Message tab.
  - *UNIX*: Select `Edit:Modify Event`.
3. To ensure that NNM posts the alarm in the Alarm Browser, make a selection in the Category field. Also make a selection in the Severity field to control the severity rating that NNM assigns to this alarm.

---

**TIP**

---

See "Creating a New Alarm Category" on page 336 for information about creating your own custom categories.

### Changing Event Configurations

1. Ensure that the prerequisites have been met (see "Prerequisites" on page 404). Then open the Event Configuration window:

- From any Alarm Browser window, select `Actions:Configure Event`.
  - From any submap, select `Options:Event Configuration`.
2. In the top half of the window, select the enterprise that supplied the MIB file that defines the SNMP event that you wish to configure. In the bottom half of the window, select the specific SNMP event:
    - *Windows*: Select `Edit:Events->Copy`, then select the `Event Message` tab.
    - *UNIX*: Select `Edit:Copy Event`.
  3. Type in a name for the new configuration that is meaningful to you.

---

**TIP**

Copy the original name and add a meaningful extension so that both events remain together in the event list.

---

4. Change the source so that your customization only applies to specific host names or IP addresses.
5. To ensure that NNM posts the alarm in the Alarm Browser, make a selection in the `Category` field. Also make a selection in the `Severity` field to control the severity rating that NNM assigns to this alarm.

---

**TIP**

See “Creating a New Alarm Category” on page 336 for information about creating your own custom categories.

---

6. To enhance the text of the alarm message so that it is more meaningful to your team, in the `Event Log Message` field, type in the new information. Follow the guidelines in “Variables and Special Characters Allowed” on page 418.

For more information, see NNM’s online help from within the `Event Configuration` window.

The customizations that you make to event configurations are written to the `trapd.conf` file. Refer to the `trapd.conf` reference page in NNM’s online help (or the UNIX manpage) for more information.

**Example Custom Alarm Message** To configure an event to post the following entry in the Alarm Browser:

```
Inconsistent subnet mask 255.0.0.0 on interface lan0,should be 255.255.248.0
```

The entry in the Event Log Message field would be:

```
Inconsistent subnet mask $9 on interface $7,should be $10
```

## Defining Automatic Actions for Events

You can define actions for NNM to perform automatically whenever a specific trap (event) is received. Follow the directions in NNM's online help available from within the Event Configuration window.

---

### NOTE

By default, NNM only performs commands that are trusted commands. If the command you specify is not in the `trustedCmds.conf` directory, then NNM generates an event and the action is not executed. See the *ovactiond* reference page (or the UNIX manpage) for more information. Trusted commands are listed in the following directory.

- *Windows:* `install_dir\conf\trustedCmds.conf`
- *UNIX:* `$OV_CONF/trustedCmds.conf`

- 
1. Ensure that the prerequisites have been met (see “Prerequisites” on page 404). Then open the Event Configuration window:
    - From any Alarm Browser window, select `Actions:Configure Event`.
    - From any submap, select `Options:Event Configuration`.
  2. In the top half of the window, select the enterprise that supplied the MIB file that establishes the SNMP trap that you wish to configure. In the bottom half of the window, select the specific SNMP trap:
    - *Windows:* Select `Edit:Events->Copy (or Modify)`, then select the `Additional Actions` tab.
    - *UNIX:* Select `Edit:Copy Event (or Modify Event)`.
  3. Type in a name for the new configuration that is meaningful to you.

---

**TIP**

---

Copy the original name and add a meaningful extension so that both events remain together in the event list.

4. Change the source so that your customization only applies to specific host names or IP addresses.
5. To configure NNM to take action automatically whenever this event is received by the Alarm Browser, in the Command for Automatic Action field, type in your instructions using “Variables and Special Characters Allowed” on page 418.

For more information, see NNM’s online help from within the Event Configuration windows.

The customizations that you make to event configurations are written to the `trapd.conf` file. Refer to the `trapd.conf` reference page in NNM’s online help (or the UNIX manpage) for more information.

### Example Commands for Automatic Action

Any program invoked from the Automatic Actions field must run to completion with no user input.

---

**NOTE**

---

By default, the following examples are not in the `trustedCmds.conf` directory. As root or administrator, you must add to the `trustedCmds.conf` directory any commands that you want to execute as automatic actions. You cannot use special characters such as `&` and `|` in your automatic action commands. See the `ovactiond` reference page (or the UNIX manpage) for more information. Trusted commands are listed in the following directory.

- *Windows:* `install_dir\conf\trustedCmds.conf`
- *UNIX:* `$OV_CONF/trustedCmds.conf`

---

**Example 1: Send a message to a pager** You can specify a paging command as an automatic action. Refer to the “Configuration: Recommended Paging Software” section of the NNM Release Notes for specific information about HP’s recommended paging software.



The following is general information about scripting an automatic action to launch pager software.

1. Install and test the pager software.
2. Create a batch file that invokes the pager software and place it in the following directory. The program invoked by the batch file must run to completion with no user input:

*Windows:* `install_dir\bin`

*UNIX:* `$OV_BIN/`

3. Identify the event that should initiate a pager action, as follows:  
Select `Options:Event Configuration`. The Event Configuration dialog box appears. Select `Edit:Events->New` to define a new event for the selected node.
4. In the `Command for Automatic Actions (optional)` field, enter the name of the batch file.

---

**NOTE**

*Windows only:* If you need to use the backslash character (for example in the batch file's path), use `"\"` (double backslash).

5. You can generate a test event using `SNMPNOTIFY` on another system to trigger this event and verify that the pager message was successfully sent.

**Example 2: Pop-up Window Message** This example pops up a message on the management station's monitor.

Specify the following in the `Popup Window Message` field:

**An authentication failure at IPaddress:\$3, community:\$4**

The output will be:

```
Authentication failure at IPaddress:15.2.77.99,  
community:admin
```

**Example 3: Set an Object's Status Color** This example assumes that you have installed the `contrib` files (see "The `contrib` Directory" on page 56). Type the following command, all on one line, in the `Command for Automatic Action` field. This event status color change will be overwritten by the `ipmap` process if the object is managed by HP

OpenView. It is suggested that a symbol in the object's submap be modified instead. See the *trapd.conf* reference page (or the UNIX manpage) for a discussion of the parameters used in an `snmptrap` command to change the status color of an object on the map.

*Windows:* If you need to use the backslash character (for example in the batch file's path), use “\\” (double backslash). Note the path statement is case-sensitive.

```
\\install_dir\contrib\NNM\setStatus\setStatus $A Testing  
UNIX:
```

```
/usr/OV/contrib/NNM/setStatus/setStatus $A Testing
```

**Example 4: Send an Audible Alert** This example assumes that you have installed the `contrib` files (see “The `contrib` Directory” on page 56). Type the following command, all on one line, in the `Command for Automatic Action` field. This event will only be heard if an `xnmevents` process is running. See the *trapd.conf* reference page (or the UNIX manpage) for more information.

*Windows:* If you need to use the backslash character (for example in the batch file's path), use “\\” (double backslash).

```
\\install_dir\contrib\NNM\ringBell\ringBell $A "User  
Alert"
```

```
UNIX:
```

```
/usr/OV/contrib/NNM/ringBell/ringBell $A "User Alert"
```

## Variables and Special Characters Allowed

When customizing event configuration's `Event Log Message`, `Popup Window Message` (in UNIX called `Pop-up Notification`), and `Command for Automatic Action` fields, use special “\$ variables” to present data that was received with the event. You may enter several special characters to give you control of the formatted output. The format can include the following standard C `printf` formats, which will be converted to their ASCII equivalent. (See also NNM's online help available from the `Event Configuration` window for more information.)

Note that nonprintable characters are converted to their `\ooo` equivalent for display in the event browser or when passed to the operator-initiated (manual) actions. The two exceptions are that a tab is displayed as `\t` in the event browser and as spaces in pop-up messages. A newline is

displayed as `\n` in the event browser and as a new line in pop-up messages. All nonprintable characters are passed unconverted to automatic actions executed by `ovactiond`.

### Special Characters

<code>\a</code>	Alert (bell) character
<code>\b</code>	Backspace
<code>\f</code>	Formfeed
<code>\n</code>	Newline
<code>\r</code>	Carriage return
<code>\t</code>	Horizontal tab
<code>\v</code>	Vertical tab
<code>\\</code>	Backslash
<code>\ooo</code>	Octal number, first character must be 0 or 1 and second and third numbers must be 0-7
<code>\xhh</code>	Hex number, both <i>hh</i> characters must be [0-9][a-f][A-F]

You can also include information from the incoming event by using the `[$arg]` format specification.

### Special Information Variables

You can also include information from the incoming event by using the `[$arg]` format specification. The following `$` variables are valid regardless of the type of event (SNMPv1, SNMPv2C, CMIP, GENERIC):

<code>[\$arg]</code>	
<code>\$x</code>	Print the date the event was received using the local date representation ( <i>UNIX only</i> : see the <code>%x</code> option in the <i>strftime</i> manpage).
<code>\$X</code>	Print the time the event was received using the local time representation ( <i>UNIX only</i> : see the <code>%X</code> option in <i>strftime</i> manpage).
<code>\$@</code>	Print the time the event was received as a number of seconds since the Epoch (Jan 1, 1970) using the <code>time_t</code> representation.

\$O	Print the name (object identifier) of the received event.
\$o	Print the name (object identifier) of the received event as a string of numbers.
\$V	Print the event type, based on the transport the event was received. Currently supported types are <code>SNMPv1</code> , <code>SNMPv2C</code> , <code>CMIP</code> , and <code>GENERIC</code> .
\$r	Print the implied source of the event in textual form. This may not be the true source of the event if the true source is proxying for another source, such as when a monitoring application running locally is reporting information about a remote node.
\$ar	Same as <code>\$r</code> except print the IP address instead of the name.
\$R	Print the true source of the event in textual form. This value is inferred via the transport mechanism which delivered the event. If the event was forwarded, this will display the address of the remote event framework (that is, remote <code>pmf</code> 's system).
\$aR	Same as <code>\$R</code> except print the IP address instead of the name.
\$c	Print the category in which the event belongs.
\$s	Print the severity of the event.
\$N	Print the name (textual alias) of the event format specification used to format the event, as defined in <code>trapd.conf</code> .
\$F	Print the textual name of the remote event framework's system (that is, remote <code>pmf</code> 's system) if this event was forwarded; otherwise, print the local system's name.
\$U	Universally Unique Identifier (UUID) of the current event in the form of a long string of alphanumeric characters.
\$\$	Print the <code>\$</code> character.

## Sequential Attribute Variables

The following \$ variables are used to access the sequential attributes that were received with the event. Each event has attributes associated with it (possibly none). They are accessed using the \$*n* notation, where *n* is the positional attribute, with 1 being the first possible attribute. The printing format is based on the ASN.1 type of the attribute. These attributes are equivalent to the variable bindings in an SNMP trap.

`[$arg]`

<code>\$#</code>	Print the number of attributes in the event.
<code>\$*</code>	Print all the attributes as <code>[seq] name (type): value</code> strings, where <code>seq</code> is the attribute sequence number.
<code>\$n</code>	Print the <i>n</i> th attribute as a <i>value</i> string.
<code>\$-n</code>	Print the <i>n</i> th attribute as a <i>name (type): value</i> string.
<code>+\$n</code>	Print the <i>n</i> th attribute as a <i>name: value</i> string.
<code>\$&gt;n</code>	Print all attributes greater than <i>n</i> as <i>value</i> strings; useful for printing a variable number of arguments. <code>\$&gt;0</code> is equivalent to <code>\$*</code> without sequence numbers, names, or types.
<code>\$&gt;-n</code>	Print all attributes greater than <i>n</i> as <code>[seq] name (type): value</code> strings.
<code>\$&gt;+n</code>	Print all variables greater than <i>n</i> as <i>name: value</i> strings.

## SNMP-Specific Variables for Traps

The following variables are valid only for events created from SNMPv1 or SNMPv2 traps:

<code>[\$arg]</code>	
<code>\$C</code>	Print the trap community string.
<code>\$E</code>	Print the trap enterprise as a text string, if possible; otherwise, as in the <code>\$e</code> arg below.
<code>\$e</code>	Print the trap enterprise as an object ID string of numbers.
<code>\$A</code>	Print the trap agent address as defined in the trap protocol data unit (PDU). Note that this may be different from the agent that actually sent the event. If the name server knows about this node, the node name will be printed; otherwise, the IP address will be printed.
<code>\$aA</code>	Same as <code>\$a</code> except print the IP address instead of the name.
<code>\$G</code>	Print the trap's generic-trap number.
<code>\$S</code>	Print the trap's specific-trap number.
<code>\$T</code>	Print the trap's <code>sysUpTime</code> time-stamp. This is the remote machine's time in hundredths of a second between the last reinitialization of the device and the generation of the trap; it is <i>not</i> the time the event was received.

### Obsoleted/Reserved Variables

The following variables are reserved or obsoleted. Their use may cause unpredictable results, and should not be used:

<code>[\$arg]</code>	
<code>\$L</code>	Obsolete, do not use.
<code>\$i</code>	Do not use.
<code>\$I</code>	Do not use.
<code>\$t</code>	Do not use.
<code>\$=</code>	Do not use.

## Defining Additional Actions

---

**NOTE**

This feature is not available through NNM's web interface. It is available on the management station and from remote consoles.

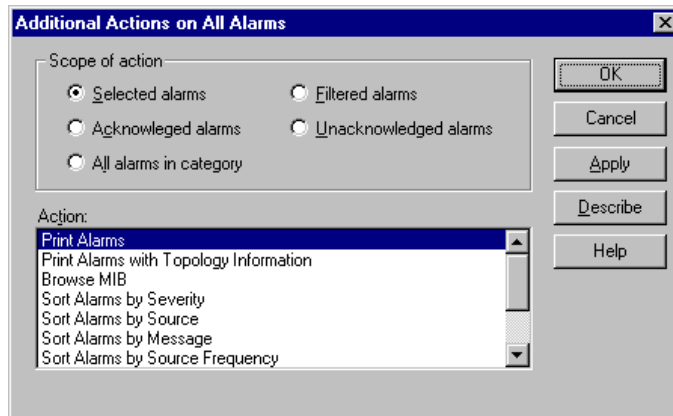
---

You can configure NNM to provide easy access to batch files or executable files that your team commonly uses. An example might be to forward the event to a trouble ticket system, to send email regarding the event, or to archive the event. You provide the programs for the additional action.

The Additional Actions window is accessed from any Alarm Browser window under `Actions:Additional Actions`.

**Figure 12-10**

### Additional Actions



*Additional Actions* are not the same as *Automatic Actions* for events. The difference between an Additional Action and an Automatic Action is that the Automatic Action automatically runs when the event is received; whereas the Additional Action is run only at the operator's choosing. The Automatic Action is tied to a particular event, whereas an Additional Action can be run on any event.

Although the additional actions list is accessed through any Alarm Browser window, you modify the list of possible additional actions through the `Event Configuration` window.

## Event Configuration Overview

To display the Event Configuration window:

- From any Alarm Browser window, select `Actions:Configure Event`.
- From any submap, select `Options:Event Configuration`.

To configure an additional action:

- *Windows*: Select `Edit:Additional Actions`.
- *UNIX*: Select `Edit:Configure->Additional Actions`.

See NNM's online help available within the Additional Actions dialog box for more information. For an explanation of how to write an additional action, see the *trapd.conf* reference page in NNM's online help (or the UNIX manpage).



---

## Using the MIB Application Builder

The MIB Application Builder application of Network Node Manager enables you to set up easy access to MIB objects that your team commonly uses. The application that you create displays real-time gathered MIB instances. You select the manner in which the information should be presented: form, table, or graph. You set up access to your application through NNM's map menu structure.

For examples of what is possible, see the standard applications provided in the map menu structure under:

- Configuration:System Information  
This example shows real-time instances of six MIB objects in *form* format.
- Fault:Test IP/TCP/SNMP  
This example shows real-time instances of four MIB objects in *table* format.
- Performance:Network Polling Statistics  
This example shows real-time instances of six MIB objects in *graph* format. You specify the interval between polling instances.

The MIB Application Builder is helpful when you want to monitor specific MIB objects on an as-needed basis. By integrating your MIB application in the menu bar, you have a convenient way to monitor multivendor network devices. For example, you could create multiple MIB applications under NNM's Performance menu that access the vendor-specific MIBs supplied by each router vendor:

```
Performance:Routers->HP
Performance:Routers->Cisco
Performance:Routers->Wellfleet
Performance:Routers->Proteon
```

---

### TIP

Use NNM's Data Collection & Thresholds application (see page 429) if you want to automate the collection, rather than use it on an as-needed basis.

---

## Prerequisites

- Make sure the enterprise-specific MIB you will use to build your MIB application is loaded in the MIB database on the management station and the same version is on the device itself. (The `Options:Load/Unload MIBs:SNMP` dialog box lists all MIB modules currently loaded in the MIB database.) If the enterprise-specific MIB is not loaded, see “Loading MIBs in the MIB Database” earlier in this chapter.
- Make sure you understand the enterprise-specific MIB. To build useful applications, you need to understand the definitions of the MIB objects and what they do.

Most device vendors include documentation about their enterprise-specific MIB with their product. This documentation typically describes the MIB objects and what the objects do. In addition, some vendors include overview information describing strategies for managing their specific devices. The vendor documentation can give you the conceptual understanding that you need to build an effective application.

Note that enterprise-specific MIBs are device-specific. For example, if you build an application for a Cisco router using a Cisco MIB, you probably cannot use that application on a Wellfleet router.

---

### NOTE

---

Familiarize yourself with the MIBs by using the `Tools:SNMP MIB Browser` operation.

- Design your application. Consider the following:
  - Which objects do you want to include in your application?
  - Which objects make sense to group?
- Should you graph this information or put it in a table or form? You can choose from one of three presentation styles -- Form, Table, or Graph.
  - `Form`. Use this option to display MIB information that is not part of a table and does not change very often (for example, system description).

The information associated with each MIB object is displayed one MIB object per line with the MIB value to the right of the MIB label. Any type of MIB object is allowed.

- **Table.** Use this option to display MIB information that constructs a logical table in the MIB (for example, a routing table).

The MIB objects are laid out horizontally, making up the columns of the table. Each entry in the MIB table makes up the rows of the application table. Any type of MIB object is allowed.

---

**NOTE**

---

You cannot create table applications which consist of MIB fields from multiple MIB tables.

- **Graph.** Use this option to display numeric MIB information that fluctuates over time (for example, interface statistics). Only numeric MIB values are allowed.
- **Plan the menu structure.** For example, if you have multiple devices from different vendors, you may want to organize the menu structure such that you have all MIB applications for routers grouped under one menu.

## Procedure Tips

Refer to NNM's online help available within the MIB Application Builder window for information about how to add, modify, or delete applications.

When you build your applications, keep the following in mind:

- You must provide a unique name for the MIB application.
- You can perform MIB operations on one or more MIB objects at a time.
- **Menu path.** Decide which NNM menu your application belongs under, such as Performance, Fault, Configuration, or Tools. Use the character sequence -> to separate menu items in the menu path field.

If the menu path field is blank, the application is created and automatically registered; however, the application will not appear in the menu bar. This is useful for creating applications to be used as the action for an executable symbol.

- You can specify selection rules for your application. Selection rules determine when a menu item is unavailable and when it is available. They are based upon defined capabilities of the nodes for which this application is compatible. If the application is registered in the menu bar, the menu item for this application is only accessible if the capabilities of the selected nodes match those defined in this selection rule; for example, `isSNMPSupported`. See NNM's online help within the MIB Application Builder window for more information.
- You can enter help text for your application. For example, you may want to give a general description of the MIB operation and how to interpret the information that is obtained. When you create the MIB operation menu item, the program automatically adds an entry under:
  - *Windows*: Help:On Application.
  - *UNIX*: Help:Misc->Function and in the dialog box help.

---

## Data Collection & Thresholds

This section describes how to use NNM's Data Collections & Thresholds application to do the following:

- Collect MIB data from network nodes automatically at regularly scheduled intervals.
- Store the collected MIB data in a file.
- Set threshold monitoring on critical devices.

See page 451 for information about viewing the data after it is collected.

---

### TIP

You cannot configure Data Collections & Thresholds from NNM's web interface. However, the web-based Alarm Browser displays the alarms resulting from your configurations.

---

### Prerequisites

- The node for which you want to collect data must support SNMP version 1 or version 2c.
- Make sure the correct version of the enterprise-specific MIB that you want to use to collect data is loaded into the MIB database on the management station. (The `Options:Load/Unload MIBs:SNMP` menu item lists all MIB modules currently in the MIB description file.) If the enterprise-specific MIB is not loaded, see "Loading MIBs in the MIB Database" on page 395.
- You can collect data or monitor thresholds on numeric MIB objects; that is, the MIB objects defined as `Counter`, `Gauge`, `INTEGER`, `IpAddress`, `Counter64`, or `TimeTicks`.
- You can collect data on MIB objects with a `String` data type; that is, monitor MIB objects such as `sysContact`.
- To configure your system to collect MIB data, you need to understand the definitions of the MIB objects and what they do.

Most device vendors include documentation about their enterprise-specific MIB with their product. This documentation typically describes the MIB object and what the object means. In addition, some vendors include overview information describing strategies for how to manage their specific device. The vendor documentation can give you the conceptual understanding you need to configure your system to collect MIB data.

- Decide if it would be useful to collect the results of a MIB expression (MIB objects combined into a mathematical formula) rather than individual MIB objects. If you want to use a MIB expression, see “Creating and Using MIB Expressions” on page 442 for information about writing and using MIB expressions.
- Identify the nodes on which you will collect data. For example, for a network segment, you may want to collect data on the router, the hubs, and the key file server.
- Identify how often you want to collect data.
- Ensure the management station has enough room to store the data. NNM automatically collects data by default (select Options: Data Collection and Thresholds to see the list of current data collections). The amount of disk space required depends upon how much data you collect. For an example, say that you want to collect data for 10 MIB objects on 50 nodes, each with an average of 8 instances of these objects. If you choose a collection interval of half an hour, the collection rate will be  $10 \times 50 \times 8 \times 2 = 8000$  values/hour. Since each value collected consumes 24 bytes of disk space, the disk consumption rate is  $8000 \times 24 = 192,000$  bytes/hour.

*Windows:* Data is stored in the `install_dir\databases\snmpCollect` directory.

*UNIX:* Data is stored in the `$OV_DB/snmpCollect` directory. If you need more disk space, symbolically link to another file system.

---

**TIP**

*For HP-UX systems only:* You can expand the file system with Logical Volume Manager. Extend the logical volume, then use `extendfs` (hfs) or `fsadm` (vxfs).

---

## Procedure Tips

To access NNM's Data Collection & Thresholds application, from any submap, select `Options:Data Collection & Thresholds:SNMP`. Refer to NNM's online help available from the Data Collection & Thresholds window for directions and additional information.

Select the MIB object or expression to be used for data collection. The `Label` field (circled in the following figure) in the MIB selection window determines the file name where the collected data will be stored. (Make sure the label conforms with file naming conventions for your operating system.) The label defaults to the last component of the MIB mnemonic name.





**Figure 12-11 Data Collection Configuration**

1. Use Edit:Modify to reconfigure an existing MIB Object or MIB expression for collection. or use Edit:New to specify a New MIB Object or expression to collect upon (GET).

2. Select the new MIB object.

This field determines the name of the file in which the collected data will be stored.

3. Configure the collection, specify:
  - \* Whether or not to store the data
  - \* Devices involved
  - \* SNMP instances to be collected upon
  - \* Collection Node Filter (if used)
  - \* Polling interval
  - \* Threshold Parameters (if applicable)

Then, configure the collection and designate the devices to be monitored.

You specify the devices to be monitored by IP host name, `sysObjectID`, IP address, IP address wildcard (such as `15.21.*.*`), and/or (*NNM for Windows only*) IPX address.

If you have a large number of nodes to specify, you can list them in an ASCII file and simply place the full path name to the file in the `List of Collection Sources` field.

As a convenience, you can select sources on any submap, then click the [Add From Map] button to populate the source list.

You can use a filter to specify the list of objects. You may use any filter defined in your filters file. See *A Guide to Scalability and Distribution* for information about writing, testing, and implementing a filter.

---

**TIP**

To verify that your filter identifies the sources you wish to monitor, use the following steps to preview the list:

1. At the command line, type:

```
ovfiltertest -f filterName > outputFileName
```

2. Verify that the generated list includes all the hostnames that you intended.

---

With wildcard IP addresses:

- All nodes or devices with IP addresses matching the wildcarded address are collected upon. (You cannot use wildcards on IPX addresses.)
- You can further limit a wildcard list by sorting upon specific `sysObjectIDs`. If you want to specify multiple `sysObjectIDs`, separate them with a comma. The `sysObjectID` itself may be wildcarded. Only nodes or devices matching both the IP wildcard and the `sysObjectID` are collected upon. (See “Unique Properties of the SNMP MIB Object `sysObjectID`” on page 447 for information about using `sysObjectID`.)

---

**NOTE**

You *must* save *any* changes for the new configuration to take effect. For example, if you suspend collection on a MIB object by selecting `Actions:Suspend Collection`, the `Status` column in the selection list indicates `Suspended`. However, even though the `Status` column displays a `Suspended` status, this change does not take effect until you select `File:Save`.

---

The data collection configuration is reanalyzed by `snmpCollect` each time you select `File:Save`. If you used IP wildcards, and you have a large number of managed nodes, there may be a significant delay before collections resume.

Counters are computed by calculating the change per second in the sampled values. Gauges and integers store the actual sampled value.

To view the collected data, open the `Data Collection & Thresholds` window, highlight the specific collection configuration in the list, and then select `Actions:Show Data`.

---

**TIP**

SNMP polling requests using the `Options>Data Collection & Thresholds:SNMP` menu item increase the amount of network traffic. If this becomes a problem:

- Change the status for particular collections to `Suspended` from within the `Data Collection & Thresholds` dialog box.
  - Modify the details of your custom data collection by either deleting nodes, deleting instances, excluding nodes, or changing polling intervals.
- 

## Defining Thresholds for Monitored MIB Objects

Thresholds let you take a proactive approach to network management. You can manually define thresholds for specific MIB objects that you are monitoring, or have NNM calculate statistical thresholds for you. You can configure events to be posted in the Alarm Browser and/or automatic actions to be taken when the specified thresholds are exceeded. Once you have established a threshold and configured a threshold event, you can

use the web Reporting interface to create threshold violation reports. See NNM's online help for more information about the web Reporting interface.

If setting fixed thresholds, first determine what would constitute reasonable thresholds. To decide on a threshold value, you need to know what is normal and what is out of range. Only you can decide what is normal behavior for a device on your network. Generally, HP recommends that you collect information about a device throughout one complete business cycle before determining the normal high/low range. Consider collecting values such as error rates, retry limits, collisions, throughput, relation rates, and many more. Explore the MIB being used to determine which information to collect.

As soon as you install NNM, data collections on certain MIB objects are started. (You can see which objects are being collected from which devices by selecting `Options>Data Collection and Thresholds`.) This data is available for automatically setting thresholds based upon deviation from normal values. NNM uses the following process to establish statistical thresholds based upon the data being collected:

1. NNM gathers data as configured in Data Collection & Thresholds (the `snmpCollect` background process collects, compares, and stores requested SNMP MIB values).
2. NNM exports the collected *trend* data to the data warehouse every hour. See the *request\_list* reference page in NNM's online help (or the UNIX manpage) for information about when NNM runs export tasks.
3. The data warehouse computes, for each instance of each configured collection:
  - The number of data points gathered in the past four days
  - The data summary
  - The sum of the squares for each collection

This information, from the previous four days, is written to the `snmpColStats.txt` file. This file is recreated each time data is exported to the data warehouse. A notification event is sent to `snmpCollect`.

These data points are sorted into three time periods, 8 p.m.-5 p.m., 5 p.m.-8 a.m., and weekends. See the *statTimeRanges.conf* reference page in NNM's online help (or the UNIX manpage) for information about controlling the definitions of the time periods.

4. When using automatic statistical threshold generation for a data collection configuration, `snmpCollect` reads `snmpColStats.txt` and calculates, one threshold standard deviation for each time period (bucket) for each data collection instance. Every time the `snmpColStats.txt` file is read, NNM compares the standard deviation calculations to the standard deviation that you specified. An alarm is generated if the current standard deviation exceeds the threshold standard deviation. See the *snmpCollect* reference page in NNM's online help (or the UNIX manpage) for more information.

Multiple thresholds are allowed per node, one for each instance; for example, one threshold set for a primary hard drive, and a different threshold set for the secondary hard drive.

---

**CAUTION**

If you are using `ovdwtrend -trim` or `-delete` options to manage the volume of your collected data, ensure that your schedule allows sufficient data history for the statistical threshold feature. See the *ovdwtrend* reference page in NNM's online help (or the UNIX manpage) for information.

---

**Prerequisites**

- Configure the data collection to which the threshold applies (see “Data Collection & Thresholds” on page 429).
- You can monitor thresholds on numeric MIB objects; that is, the MIB objects defined as `Counter`, `Gauge`, `INTEGER`, `IpAddress`, `Counter64`, or `TimeTicks`. You can also set thresholds on MIB expressions (see “Creating and Using MIB Expressions” on page 442).

**Procedure Tips**

Access the collection configuration dialog box of the configured Data Collection you have been using to establish the normal operating range of your device. Select *one* of the following menu items from the `Option` menu next to the `Collection Mode` label:

- Don't Store, Check Thresholds
- Store, Check Thresholds  
(required if you want to use the statistical threshold feature)

The Threshold and Rearm fields are now available.

**Figure 12-12** Threshold Area of Data Collection Window

Threshold Parameters

Threshold

Both Statistical and Fixed

Fixed Threshold: > 25

Statistical Threshold: Above 3 Standard Deviation

For: 1 Consecutive Samples

Rearm

Either Statistical or Fixed

Fixed Rearm: <= 5

Statistical Rearm: Above 0 Standard Deviation

For: 4 Consecutive Samples

Rearm Value Type

Percent Of Threshold

Absolute

Threshold Event Num: 58720263

Configure Threshold Event...

Configure Rearm Event...

To set up fixed-value threshold monitoring, select “Fixed” or “Both Statistical and Fixed” mode, specify a threshold value, and specify the number of consecutive samples required to violate the threshold. To set up fixed-value rearm monitoring, specify a fixed rearm value or percent, and the number of consecutive samples required to rearm the threshold.

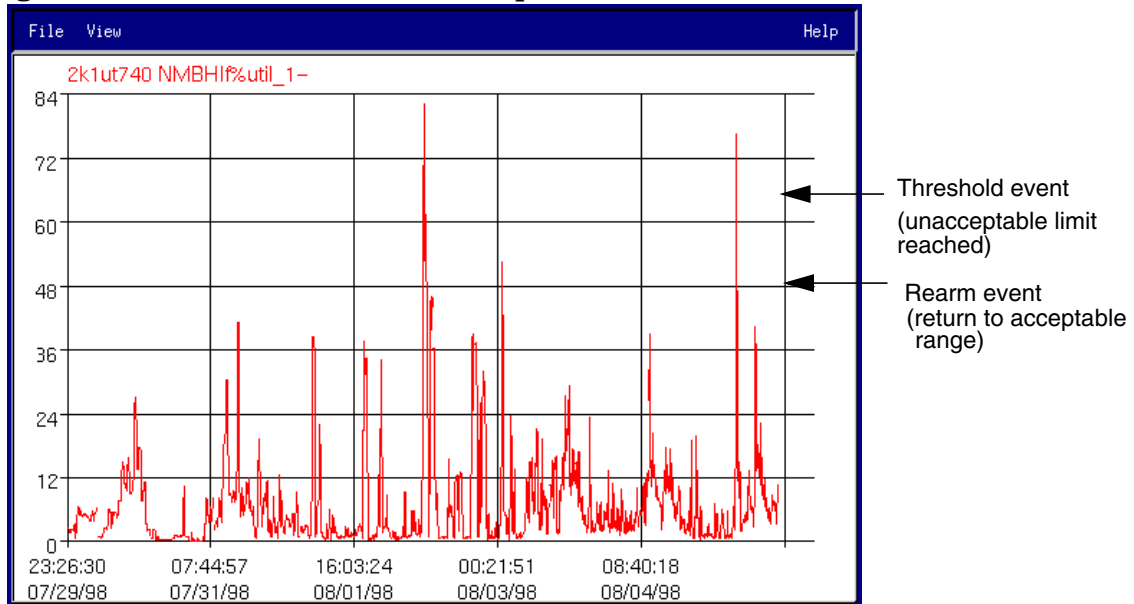
To set up automatic statistical threshold monitoring, select “Statistical” or “Both Statistical and Fixed” mode, specify the standard deviation, and specify the number of consecutive samples required to violate the threshold. To set up automatic statistical rearm monitoring, select “Both Statistical and Fixed” or “Either Statistical and Fixed” mode, specify standard deviation, and specify the number of consecutive samples required to violate the threshold. NNM automatically calculates the standard deviation according to the data warehouse’s schedule for exporting trend data.

When the threshold setting is exceeded, a threshold-exceeded alarm is posted in the Alarm Browser, and any automatic actions (such as dialing a pager) that you specify in the Event Configuration are executed.

The *rearm value* controls how frequently a threshold event is generated. The *rearm* indicates that the device has returned to a non-critical state. A *rearm alarm* is posted in the Alarm Browser and any automatic

actions that you specified in the Event Configuration are executed. Another threshold event is not generated until the rearm event occurs and the collected value again reaches the threshold setting.

**Figure 12-13** Data Collection Helps Establish Realistic Thresholds



For example, suppose you want to monitor interface percent-utilization. You can set a `Threshold` of 70% and a `Rearm` value of 50%. When the data collector detects a value above 70%, a threshold alarm is generated. Another threshold alarm will be generated only if the reading drops below 50% (sending a rearm alarm) and then goes above 70% again. Note that you can also specify absolute values rather than percentages.

Optionally, specify a value for the `Consecutive Samples` field. This value specifies the number of consecutive times the `Threshold` or `Rearm` expression must be satisfied before a corresponding alarm is generated. This field lets you distinguish sustained abnormality from a brief spike in behavior.

NNM uses two preconfigured events under the `OpenView enterprise` name (.1.3.6.1.4.1.11.2.17.1.0.) as the default events for thresholds and rearms:

```
OV_DataCollectThres (58720263)
```

OV\_DataCollect\_Rearm (58720264)

The default alarm is posted in the Alarm Browser's Threshold Alarms category.

The default threshold and rearm events may not meet all of your needs. Suppose, for example, that some of the threshold values you are monitoring are more critical than others. In these cases, you can configure NNM to take special action upon receipt of alarms for a particular threshold. For example, you could configure NNM to send an email message when it receives the alarm.

The easiest way to establish your own custom threshold and rearm alarms is to click on the [Configure Threshold Event] button, and copy the default event configuration. Supply your own event name and number. NNM reserves .1.3.6.1.4.1.11.2.17.1.0.1-10000 for custom threshold/rearm events. Threshold event numbers are always odd numbers between 1 and 9999. The corresponding rearm event numbers are always equal to the corresponding threshold number plus 1.



**NOTE** If you create a custom threshold event, you must also create a matching rearm event.

**Figure 12-14 Customizing the Threshold Event**

**1** Click on the Configure Threshold Event button.

**2** Copy the default OV\_DataCollectThresh

**3** Type a new name, for example:  
OV\_DatCollectThresh\_XXX

**4** Type an odd number between 1 and 9999  
(Remember to change the number in the collection configuration dialog box to match.)

**Copy Events**

Event Name: Enterprise: OpenView

Generic Trap: Enterprise Specific

Specific Trap Number: 58720263

Description: This event is generated by HP OpenView data collector when a sampled value exceeds a preconfigured level. (See also specific event #58720264 for rearm events). The user may also configure value exceeded events with specific trap numbers in the odd range of 1 to 9999.

For example, the threshold event ID 1005 will have a rearm event ID of 1006. These rearm event IDs are automatically assigned but *not* automatically configured in NNM. If you use one of the optional threshold event numbers, you must configure events for both the threshold and rearm event IDs (use the [Configure Threshold Event] and [Configure Rearm Event] buttons).

---

**NOTE**

If you are configuring a threshold event created by the HP OpenView SNMP Data Collector, you should configure it under the OpenView enterprise name (.1.3.6.1.4.1.11.2.17.1.0.), regardless of which node the Data Collector is polling. Even if the event is for a Cisco device exceeding a threshold, the enterprise ID of the event will be OpenView, not Cisco, since the event is coming from the HP OpenView SNMP Data Collector.

---

Refer to *A Guide to Scalability and Distribution* for information about forwarding threshold events to another management station.

## Collecting and Storing Textual Data

NNM can collect MIB objects with a String data type. This data can be used to monitor device changes. For example, you can set up data collection on SysContact.

The collected string data is stored in the following locations.

- *Windows:* `install_dir\databases\snmpCollect\`
- *UNIX:* `$OV_DB/snmpCollect/`

When string data is exported to the data warehouse, only new values are stored. The "period" field shows how long the value has remained the same. String data is not aggregated into daily, Weekly, Monthly, or Yearly tables. When exporting text data to the data warehouse, the existing data is deleted from the trend database.

## Creating and Using MIB Expressions

MIB expressions are mathematical formulas comprised of MIB objects. Expressions let you derive more meaningful information about the health of your network than you could gather using individual MIB objects. MIB expressions are used by several NNM components, and are

available for your use through the Data Collection & Thresholds window. You can also create your own and add them to the list by making additions to the `mibExpr.conf` file and the `mibExpr.conf` file.

### **Default MIB Expressions**

NNM includes many default MIB expressions for your convenience. To explore the default MIB expressions, from any submap, select Options>Data collection & Thresholds, then select the following menu item:

*Windows:* Edit:MIB Object->New

*UNIX:* Edit:Add->MIB Objects...

Select the Expressions radio button. The list of predefined MIB expressions is displayed for your use. The following are a small sampling of the MIB expressions provided:

**Equation 12-1**      **If%deferred (MIB II)**  
**The percent of packets on an interface which were deferred:**  
 (packets deferred rate ÷ packets transmitted rate) × 100

**Equation 12-2**      **If%collision (MIB II)**  
**Percent of packets on an interface which collided:**  
 (collision rate ÷ packets transmitted rate) × 100

**Equation 12-3**      **If%inErrors (MIB II)**  
**Percent of input packets on an interface which had an error:**  

$$\left( \frac{\text{input error rate}}{\text{received unicast packet rate} + \text{received non-unicast packet rate}} \right) \times 100$$

**Equation 12-4**      **If%outErrors (MIB II)**  
**Percent of output packets on an interface which had an error:**  

$$\left( \frac{\text{output error rate}}{\text{transmitted unicast packet rate} + \text{transmitted non-unicast packet rate}} \right) \times 100$$

**Equation 12-5**      **If%util (MIB II)**  
**Percent of available bandwidth utilized on an interface:**  

$$\left( \frac{(\text{received byte rate} + \text{transmitted byte rate}) \times 8}{\text{interface link speed}} \right) \times 100$$

**Equation 12-6**      **IfInOctets (MIB II)**  
**Total absolute traffic on an interface or the summation of octets (bytes) sent and received on an interface:**  
 received byte rate + transmitted byte rate

**Equation 12-7**      **IfInOutPackets (MIB II)**  
**Total absolute traffic on an interface or summation of packets sent and received on an interface:**

$$\left( \frac{\text{received unicast packet rate}}{\text{received non-unicast packet rate}} + \frac{\text{transmitted unicast packet rate}}{\text{transmitted non-unicast packet rate}} \right)$$

**Equation 12-8**     **Disk%util (HP-UX MIB)**

**Percentage of disk utilization:**

$$\left( \frac{\text{blocks} - \text{free blocks}}{\text{blocks} - \text{free blocks} + \text{available blocks}} \right) \times 100$$

Simply select one from the list and configure it for data collection and/or threshold monitoring on your network's devices.

---

**TIP**

There are seven additional MIB expressions defined specifically for use with the HP-extensible agent. They are located in:

- *Windows:* `install_dir\conf\extExpr.conf`
- *UNIX:* `$OV_CONF/extExpr.conf`

Open the file to read about them.

---

**The mibExpr.conf File**

The expressions are stored in an ASCII file:

- *Windows:* `install_dir\conf\mibExpr.conf`
- *UNIX:* `$OV_CONF/mibExpr.conf`

You can modify this file or create your own expressions using the `xnmcollect -loadExpr` utility. Best practice is to create a separate file with your new or revised MIB expressions, and then merge the expressions into the `mibExpr.conf` file using the following utility from the command prompt:

```
xnmcollect -loadExpr filename
```

The `-loadExpr` checks syntax, removes duplicate expressions, and rewrites the file.

See the *mibExpr.conf* reference page in NNM's online help (or the UNIX manpage) for information about writing your own MIB expressions; and see the *xnmcollect* reference page in NNM's online help (or the UNIX manpage) for information about updating the `mibExpr.conf` file.

### The `mib.coerce` File

The `mibExpr.conf` file works in partnership with the `mib.coerce` file. Entries in the `mib.coerce` file instruct NNM to convert (coerce) a received MIB value from one data type to another (for example, from a gauge or counter to an integer) for use in the mathematical formula defined in the `mibExpr.conf` file. The `mib.coerce` file is an editable ASCII file:

- *Windows:* `install_dir\conf\mib.coerce`
- *UNIX:* `$OV_CONF/mib.coerce`

After making changes to the `mib.coerce` file, you must force NNM to acknowledge the changes; from the command line, type:

**`xnmloadmib -event`**

See the `mib.coerce` reference page in NNM's online help (or the UNIX manpage) for more information about converting MIB values from one data type to another for use in your MIB expression.

## Unique Properties of the SNMP MIB Object sysObjectID

The `sysObjectID` is an important SNMP MIB object to NNM operations. This section describes:

- What the `sysObjectID` is
- How network management operations use the `sysObjectID`

The `sysObjectID` is registered in the Internet-standard MIB-II module as `iso.org.dod.internet.mgmt.mib-2.system.sysObjectID` (1.3.6.1.2.1.1.2). The `sysObjectID` is used for administrative purposes to uniquely identify the type of SNMP agent software that is running on a given vendor's hardware. This object is different from most other MIB objects; when you query it, you get back an object identifier that describes the product. For example, when you query the `iso.org.dod.internet.mgmt.mib-2.system.sysObjectID` using the MIB Browser on an HP OpenView SNMP Agent that is running on an HP 9000 Series 700 or Sun SPARCstation, SNMP returns one of the following object identifiers:

```
iso.org.dod.internet.private.enterprises.hp.nm.system.hpsun.sparc.sun4  
iso.org.dod.internet.private.enterprises.hp.nm.system.hpsun.sparc.sun5  
iso.org.dod.internet.private.enterprises.hp.nm.system.hpux.hp9000s700
```

This result tells you how the HP 9000 Series 700 or Solaris agent is registered in the MIB. The numeric identifier for Solaris is 1.3.6.1.4.1.11.2.3.10.1.2

Note that there is no value associated with the object ID returned in `sysObjectID`. It is a unique identifier.

The `sysObjectID` includes both hardware and software information, and varies depending on the type of hardware and agent software. Some vendors have a different `sysObjectID` for every version of their agents. For example, Hewlett-Packard sells several SNMP agents, each uniquely identified. To distinguish the HP 9000 Series 800 agent from the HP 9000 Series 700 or Solaris agent, HP registered the HP 9000 Series 800 agent as:

```
iso.org.dod.internet.private.enterprises.hp.nm.system.hpux.hp9000s800
```

## Unique Properties of the SNMP MIB Object sysObjectID

netmon, a service (background process) that automatically discovers the nodes on your network, obtains the `sysObjectID` when it queries SNMP nodes. NNM operations take advantage of this special object in the following ways:

- To identify the vendor; that is, the manufacturer of the hardware.
- To identify the source of the SNMP agent, which is the type of agent software running on the hardware. For example, if you do a `GetRequest` to a Sun SPARCstation, the vendor is Sun, but the SNMP agent may be
  - An HP OpenView SNMP Agent for SunOS
  - An SCO UNIX for SunOS agent
  - Other SNMP agent software
- To determine IP topology behavior. An agent's `sysObjectID` provides hints about what type of node the agent is; for example, a gateway, a bridge, or a host.
- To determine symbol type when putting symbols for newly discovered nodes on the map; for example, a bridge, a gateway, or a hub.
- To identify the source of an event. A `sysObjectID` is sent in every event. You can use the `sysObjectID` when you configure events and data collections. The reference in the Event Configuration dialog box to enterprise ID is generally synonymous with the `sysObjectID` of the agent that generated the event.

Like NNM, you can take advantage of the `sysObjectID` in many ways, such as:

- From any submap, select `Find:Object By Attribute` to locate quickly all of a certain manufacturer's model-X when they require a software upgrade.
- If you are writing a rules-based application or script, use the `sysObjectID` as a rule index.
- On any submap, select a node and use `Edit:Object Properties` to quickly determine which SNMP agent is running on that node.
- Use the `sysObjectID` in your filter definitions: discovery, topology, map, etc.



- Use the `sysObjectID` when using the `ovtopodump` command to generate useful lists.
- Customize NNM to display a symbol that you created for a specific type of object (see Appendix C, “Changing All the Symbols for a Particular Device,” on page 623).
- If NNM displays an object on the map with a generic symbol (just the outside shape, no vendor-specific icon within the shape), you can specify the icon of your choice to be mapped to that device’s `sysObjectID` (see Appendix C, “Changing All the Symbols for a Particular Device,” on page 623).
- Specify nodes for which data should be collected, as described in the previous section.

Customizing Events: Doing It Your Way

## Unique Properties of the SNMP MIB Object sysObjectID

---

# **13**      **Using Event Data**

NNM provides several features that can assist you in analyzing the network data that has been collected:

- *Graph SNMP Data* (page 453)

Display collected SNMP data in graphs. This can be SNMP data that has been collected over time and stored in a file, or real-time values that are graphed as they are collected, or both at the same time.

- *Data Warehouse* (page 458)

Allows you to export collected historical data for use with other statistical analysis tools. Command line functions allow you to maintain the data warehouse database.

- *Web Reporting Interface*

Allows you to use data from the data warehouse to create reports using a set of predefined templates. See the *Reporting and Data Analysis* online manual and the online help for more information.

---

**NOTE**

For information about how to merge data from multiple NNM collection stations into a single database, see *A Guide to Scalability and Distribution*.

---

Ideas for utilizing your collected data:

- Anticipate the need for more network and computer resources, such as gateways and disk space.
- Detect deviations in normal activity.
- Isolate network faults and performance problems.
- Generate inventory reports.

## Graphing SNMP Data

The NNM Grapher tool allows you to graph collected MIB data from within the SNMP MIB Browser. Once the Grapher window is displayed, you can set time intervals to continue the collection and even automatically establish a configuration for continuous monitoring by selecting `File:Configure` in Data Collector.

The NNM Grapher tool also enables you to organize MIB data that was collected by the Data Collections & Thresholds feature and view it in graph form. You can graph combinations of data values in the same graph. The data values can be different instances of MIB variables or different variables for different nodes. For example, you can graph instance 1 and instance 3 of a MIB variable for one system, or you could graph instance 3 of a MIB variable for one system, and a completely different variable for another system. You can also normalize the data over configurable time periods to identify trends in the historical data.

This section describes the following:

- Graphing collected data, which includes graphing all data or graphing selected data only, and updating data that has been graphed.
- Graphing historical and real-time data in the same graph.
- Zooming in and out of a specific period of time to view the data in greater or less detail.
- Modifying line attributes in a graph.
- Printing a graph.

---

### NOTE

The Grapher tool is available in NNM on a Windows or UNIX operating system. If you are using NNM on the Web, there is a Grapher tool available in the `contrib` directory. (See page 56 for information about using programs from the `contrib` directory.)

---

## Graphing Collected Data

You can graph the data collected in the Data Collector's database by using the Performance:Graph SNMP Data->All and Performance:Graph SNMP Data->Selected Nodes menu items. You can also access the Grapher tool from within the SNMP MIB Browser or by using the `xnmgraph` command. Refer to the *xnmgraph* reference page in NNM's online help (or the UNIX manpage) for more information.

### Behavior

The Graph Selected Nodes dialog box consists of a line graph. The legend across the top of the graph identifies the currently graphed data. Time is shown on the X-axis.

The menu bar has functions that enable you to manipulate the data. The menu bar also has help entries explaining how to use the different menu options to manipulate the data.

The Graph SNMP Data operation does not update the graph with new information as the data is collected. To update the graph with the most current MIB data, select File:Update Data from the Grapher menu bar.

Refer to the online help within the graph window for additional information.

### Graphing All Data

The Performance:Graph SNMP Data->All operation enables you to view collected data in graph form without first having to build an application. This is useful when you want to browse all the data in the Data Collector's database:

- *Windows*: Data is stored in the `install_dir\databases\snmpCollect` directory.
- *UNIX*: Data is stored in the `$OV_DB/snmpCollect` directory. (If you need more disk space, symbolically link to another file system.)

---

#### TIP

*For HP-UX systems only*: You can expand the file system with Logical Volume Manager. Extend the logical volume, then use `extendfs` (hfs) or `fsadm` (vxfs).

---

Use this operation when you want to graph some data points just once, or when you want to compare different variables across different systems. For example, let's say you want to compare the load averages to the number of packets, but had not used the Application Builder (see page 425) to set up your own application for this purpose. You can easily use the Graph All operation to graph these two variables together.

---

**CAUTION**

This operation reads in *all* the collected data from the Data Collector database into its memory. Depending on the amount of data stored in the database, displaying the Graph All dialog box may take a long time.

---

### Graphing Data on Selected Nodes

The Performance:Graph SNMP Data->Selected Nodes operation enables you to view collected data on the nodes that are currently selected on the map, without having to use the Application Builder (see page 425) to set up your own application for this purpose. This is useful when you want to browse the collected data for these nodes. Use this operation when you want to graph some data points just once, or when you want to compare different variables across different systems.

For example, let's say you want to compare the load averages to the number of packets on the selected systems, but had not used the Application Builder (see page 425) to set up your own application for this purpose. You can use the Graph Selected Nodes operation to easily graph these variables together.

### Graphing Historical and Real-Time Data in the Same Graph

When you use the Grapher tool via an NNM-provided menu item, the Grapher displays both the collected data and real-time data in the same graph (for example, Performance:Network Activity).

When Grapher starts up, it does the following:

1. Reads in the data from the Data Collector's database.

When the Grapher tool first comes up, it loads all the collected data from the Data Collector database and seeds its memory with the information in the database. The resolution of the collected data is determined by what the Data Collector was configured to collect.

Once the Grapher tool is running, the tool does not interact with the data in the Data Collector database again (unless you select `File:Update Data`). The Data Collector operates in the background and continues to collect data. This data is available for you to use in future operations.

## 2. Starts to graph the real-time data.

The real-time data displayed is independent of the Data Collector. The Grapher tool processes SNMP requests based on the interval specified under `View:Time Intervals` and displays the data in the graph. Unless you save the graphed data to a file using the `File:Save As` menu item, the real-time data is lost as soon as you exit the Grapher.

## Stopping Real-Time Data Graphing

To stop real-time data graphing, from the Graph window:

- *Windows:*  
Select `Options:Time Intervals->SNMP Polling Off`.
- *UNIX:*  
Select `View:Time Intervals->SNMP Polling Off`.

## Grapher Operations

This section describes some common operations for manipulating a graph.

### Selecting Time Intervals

To select time intervals, using the left mouse button:

- Click on the left edge of the graph to page back one screen.
- Click on the right edge of the graph to page forward one screen.
- Click within the graph to center the graph around the selected point in time. Holding the mouse button down and dragging enables you to fine tune the point that you want centered.



## Zooming In and Out of a Graph

The Grapher allows you to zoom in and out of a specific period of time in order to view data in greater or lesser detail. To do so, using the right mouse button, click anywhere within the graph. An option menu appears. You can zoom in and out to different amounts of detail. You can also use the menu to page forward, page backward, or show different data points.

## Modifying a Line Attribute

To access the Line Configuration dialog box, from the Graph window:

- *Windows*: Select `Options:Line Configuration`.  
Option buttons are arranged in rows and columns. Each item in the list controls the attributes for a particular line on the graph.
- *UNIX*: Select `View:Line Configuration`.  
Option buttons are arranged in rows and columns. Each row in the list controls the attributes for a particular line on the graph.

## Printing a Graph

To Print a graph, from the Graph window:

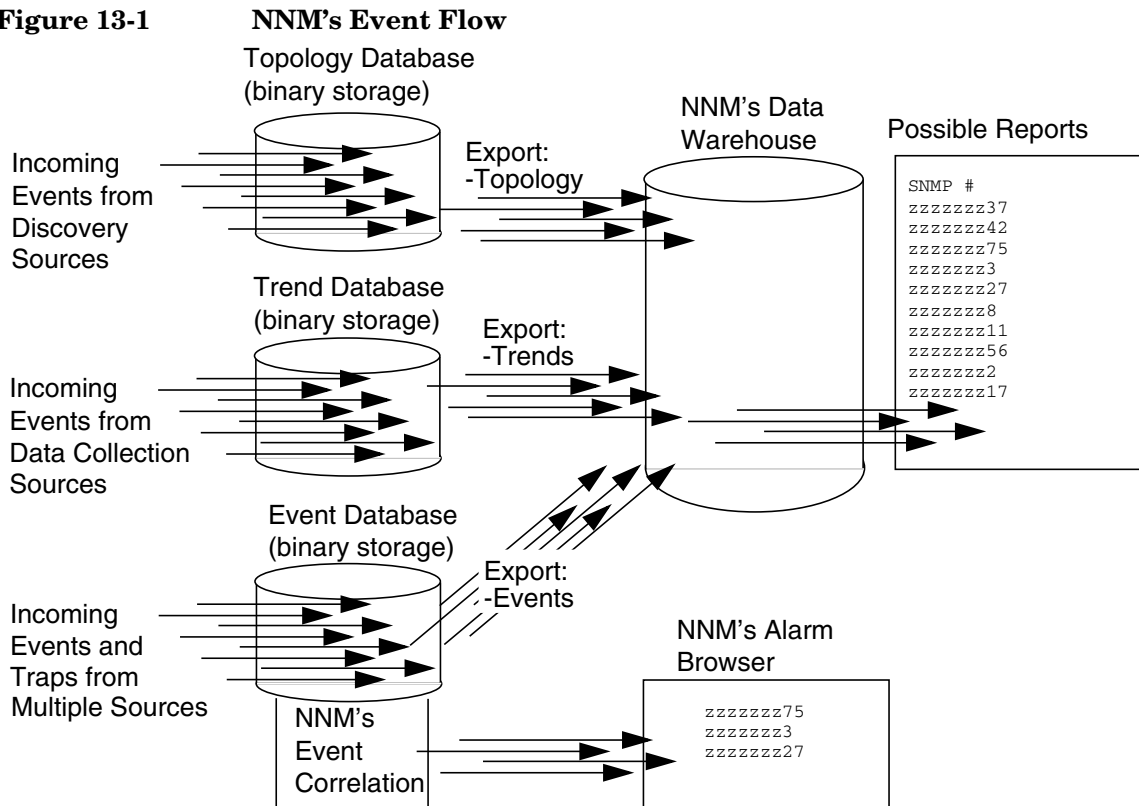
- *Windows*: Select `File:Print`. The standard Print dialog box appears. Make your selections and click [OK].
- *UNIX*: An `xwd`-compatible file is produced by the Grapher. The default Print command is configured in the Grapher print resource. The commands shown in the Print Command window are used to print that file. If the default command does not meet your needs, you can change it. Before you configure the default Print command, determine which command options will give you the best printouts. See NNM's online help for more information.

## Data Warehouse

The NNM data warehouse is a relational database (RDBMS) provided with NNM. NNM stores a *copy* of the information from the NNM operational databases into the data warehouse so you can gain an historical perspective of your network information. The data warehouse information is accessible with standard SQL statements using ODBC tools. You can also create reports from the data using the NNM web Reporting interface.

By contrast, the information in the NNM *event database* is discarded when a certain volume is reached (see page 330). Information copied to the *data warehouse* is retained for as long as you need it. You can derive reports from hourly, daily, weekly, monthly, or even yearly information. The limit to the size of the data warehouse is up to you and the allotted hard drive space for the data warehouse.

**Figure 13-1**



NNM periodically populates the data warehouse according to a schedule. See the *ovrequestd* and *request\_list* reference page in NNM's online help (or the UNIX manpage) for more information. To view the current schedule, at the command prompt, type:

**request\_list schedule**

You can also manually populate the data warehouse using the following pull-down menu items:

- Tools>DataWarehouse->Export Topology

Exports the current network topology information, replacing any topology information previously sent to the data warehouse.

- `Tools:DataWarehouse->Export Events`

Exports any event database information that is more current than the last time the events database was exported. (Remember: NNM's event database is automatically trimmed when it reaches maximum size, see page 330, so it *is possible* that events will be deleted before you do the next export.)

Events can be filtered as they are exported to the data warehouse to reduce the volume. Refer to the *ovdweventflt* reference page in NNM's online help (or the UNIX manpage) for more information. To view the list of existing export filters, at the command prompt, type:

```
ovdweventflt -display
```

- `Tools:DataWarehouse->Export Trend Data`

Exports any data in the Data Collections & Thresholds database (`snmpCollect`, see page 429) that is more recent than the last time the data was exported.

NNM's reporting feature controls how long information is stored in the data warehouse and the schedule upon which the stored information is trimmed. See NNM's online manual, *Reporting and Data Analysis*. See also the *ovdutopo*, *ovdwevent*, *ovdwtrend* reference pages in the NNM online help (or the UNIX manpages) for information about configuration and maintenance of the information stored in the data warehouse.

## Data in the Data Warehouse

You can use the data in the data warehouse with other statistical analysis programs or network management tools. You can also use the data to create reports with NNM's web Reporting interface.

See the NNM online manual, *Reporting and Data Analysis*. See also the NNM online help.

## Reporting

Reporting is enabled by default when NNM is installed. Once the report data is collected, daily and month-to-date reports for general availability, threshold violation exceptions, and network performance are available from the `Tools:Report Presenter` command or from the HP OpenView

Launcher. Daily Inventory reports are available as soon as the report data is collected. For more information about reporting see page 27 in this manual.

For more information about the Reporting feature, see the section labeled “Reporting Interface” on page 506 in this manual.

For more information about the NNM databases and the data warehouse, consult the online manual *Reporting and Data Analysis*.



---

# **14** **NNM on the Web**

This chapter describes HP OpenView's **Home Base**, used for launching Dynamic Views. It also describes HP OpenView's Java™-based web interface, and tells you how to set up, configure, and administer NNM's web applications.

This chapter will not attempt to teach you how to use all the features of HP OpenView's web interface. You should consult the NNM online help within the web interface for usage instructions.

This chapter refers to HP OpenView Application Registration Files (ARF) in several places. If you are not familiar with that file, review its functionality in the online manual *Creating and Using Registration Files*.

---

**TIP**

If you make changes or additions to the web registration files ensure that your new files are properly backed up. They are *not* included in NNM's backup scripts. See "Backup/Restore to Protect Your Investment of Time" on page 149 for more information.

---

This chapter includes:

- An overview of Home Base, a launching point for many of the Dynamic Views (page 466).
- A description of several views of discovered nodes (page 467).
- An overview of the HP OpenView web applications, including setting up the web applications and security for the web applications (page 474).
- A description of the HP OpenView Launcher, the UI integration point for the web applications, including configuration details (page 482).
- A description of the Network Presenter, the web application that displays maps and map data (page 491).
- A description of the SNMP Data Presenter, which displays the results of operations that gather data from managed SNMP nodes (page 499).



- A description of the web-based Alarm Browser, a graphical interface that presents network alarm data, including a comparison of the web-based Alarm Browser and the Alarm Browser running on the management station (page 502).
- A description of Event Correlation Services, NNM's multi-protocol event correlator (page 505).
- A description of the Reporting interface, the web application that allows you to configure and view reports (page 506).

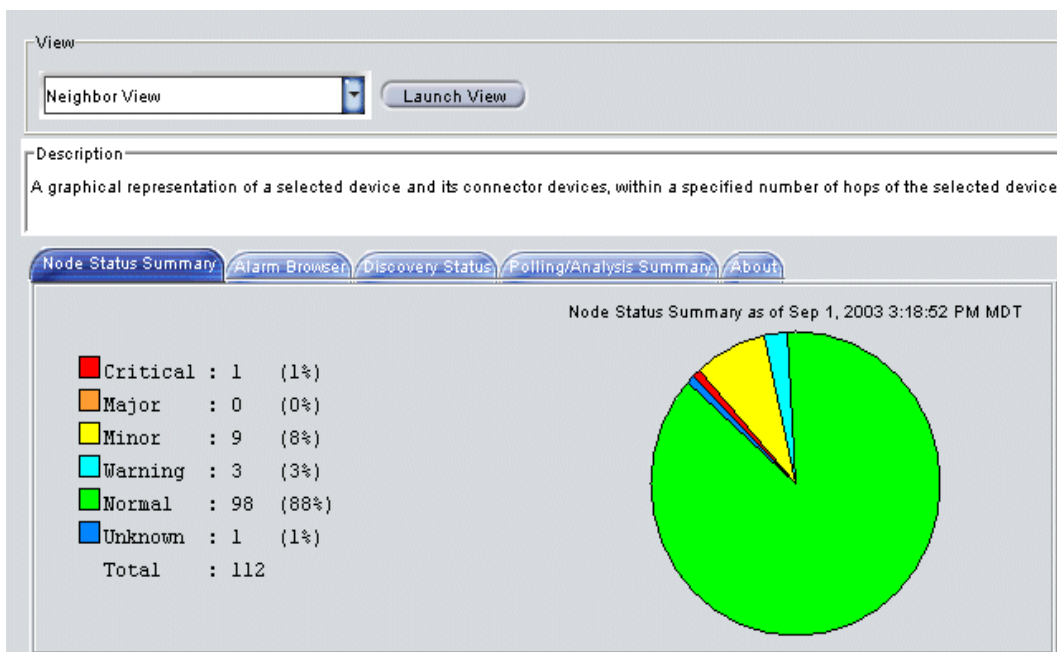
## Overview of Dynamic Views and Home Base

Dynamic Views describes the family of browser-based views whose content is created as a result of choices you make when you launch the view, and which continue to provide the most current status information available.

Home Base is a launching point for many of these Dynamic Views. In addition to launching views from Home Base, you can select tabs that cause Home Base to display additional information about your network. See Figure 14-1 on page 466 for an example of Home Base. You can access Home Base from your browser using the following URL:

`http://hostname:7510`

**Figure 14-1** Home Base



## Dynamic Views

The NNM views available to you depend on whether you purchased NNM's Starter or Advanced Edition and whether you purchased the Advanced Routing SPI along with NNM's Advanced Edition. For more information about the features included in NNM's Advanced Edition and the Advanced Routing SPI, see *Using Extended Topology*.

NNM presents many different views of discovered nodes. NNM's Neighbor, Node, Station, Internet, Network, Path, VLAN, Problem Diagnosis, HSRP, OSPF, OAD, IPv6, and Port-Address Mapping views show topology views and other information that supplement NNM's map views. Below are some examples of a few of these views. See NNM online help for additional information about each view.

### Dynamic View Security

To set up Dynamic View Security, run the `dvUsersManager.ovpl` script with no parameters. It will prompt the user to enter the username and password. See the `dvUsersManager.ovpl` reference page (or the UNIX manpage for more information).

Once the username or password has been changed from the factory default, the `dvUsersManager.ovpl` script cannot be used to change them again. To configure additional user roles and passwords, edit the following file:

- **Windows:**  
`%OV_AS%\webapps\topology\WEB-INF\dynamicViewsUsers.xml`
- **UNIX:**  
`$OV_AS/webapps/topology/WEB-INF/dynamicViewsUsers.xml`

This file contains examples of how to set up various user roles. You can create an operator role that has access to all of the Dynamic Views, but cannot access any of the configuration tools. See the *DynamicViewsUsers.xml* reference page in NNM's online help (or the UNIX manpage) for more information.

If non-ASCII characters are added to XML files, you must preserve the UTF-8 codeset for these characters.

**NOTE**

The Windows Notepad editor allows files to be saved in the UTF-8 code set. On many UNIX platforms, the iconv command can be used to translate from Shift JIS (or any other code set) into UTF-8 to make editing in a non-UTF-8 codeset simpler.

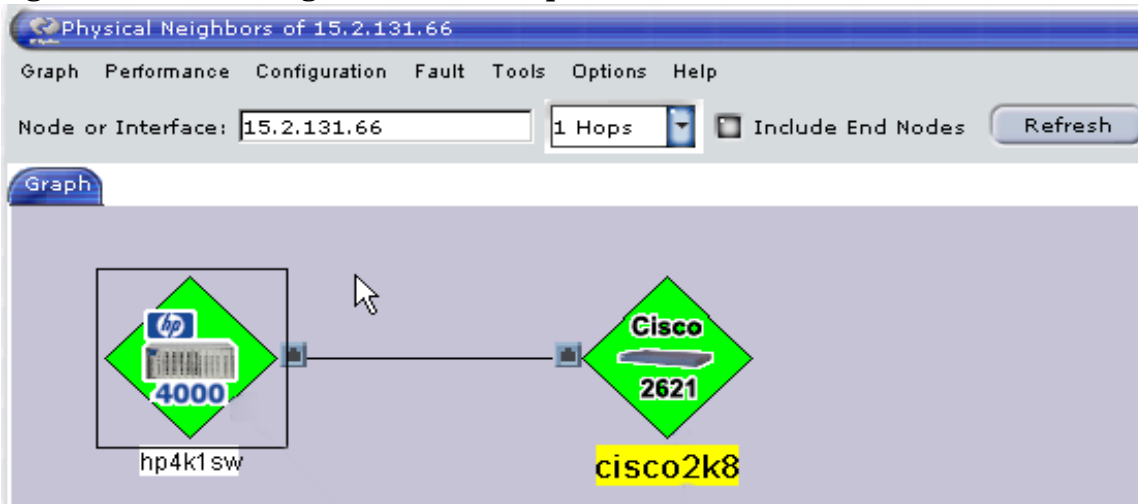
For more detailed information on setting up passwords, including how to use MD5 encryption, look for additional instructions contained in the following file:

- *Windows*: %OV\_AS%\webapps\topology\WEB-INF\web.xml
- *UNIX*: \$OV\_AS/webapps/topology/WEB-INF/web.xml

**Neighbor View**

The Neighbor view shows you a graphical representation of the selected device and the connector devices related to it, within a specified number of hops of the selected device.

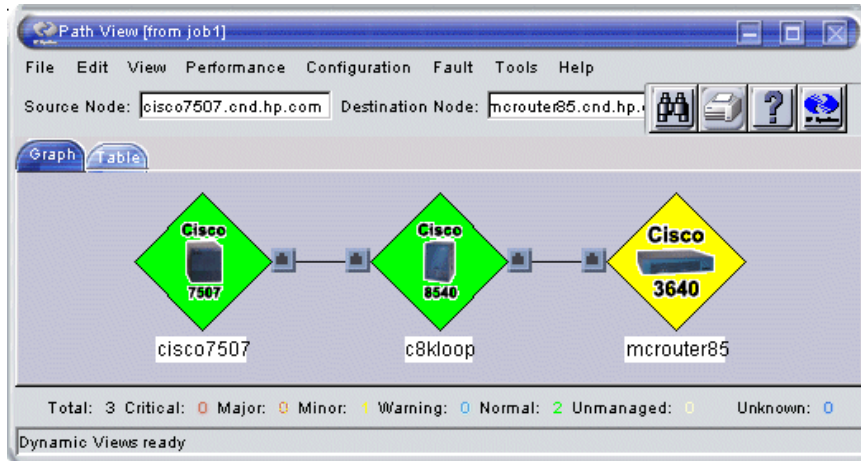
**Figure 14-2** Neighbor View Example



### Path View

The Path view shows you a graphical representation of the path between two nodes. NNM calculates the shortest path between the source and destination nodes from information contained in the topology database.

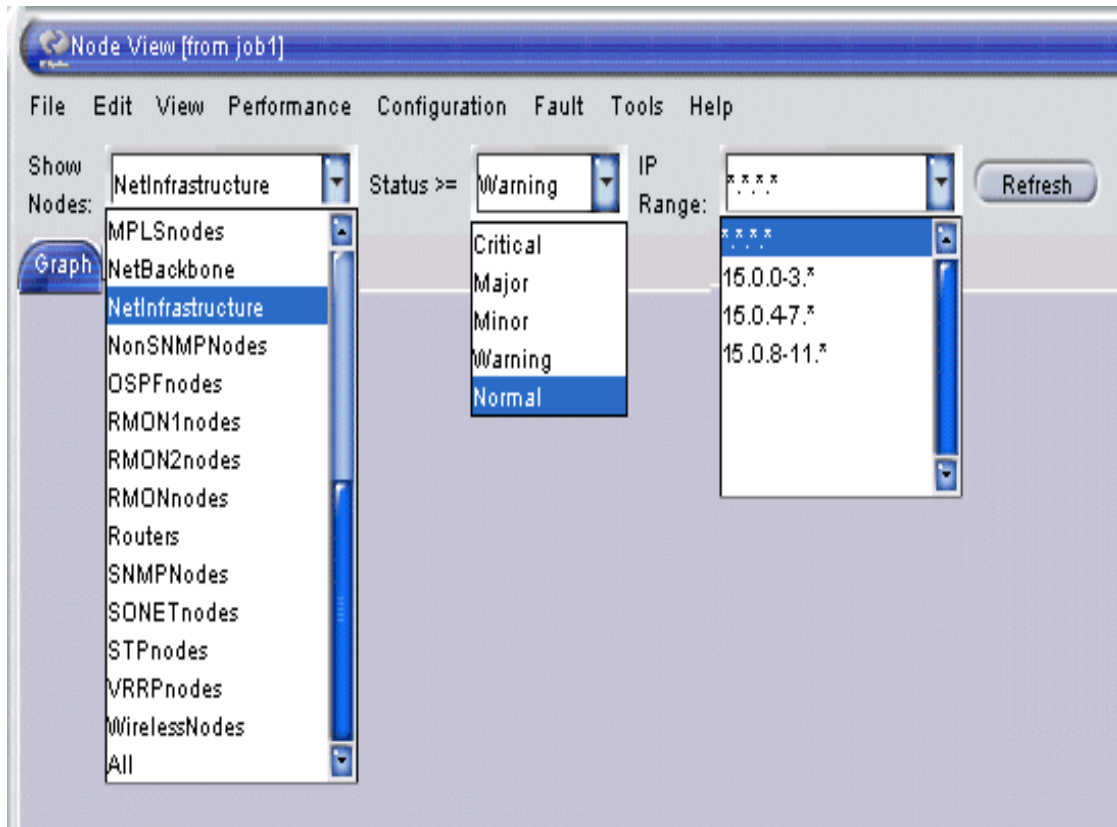
**Figure 14-3** Path View Example



### Node View

The Node view creates a visualization of a set of nodes and shows how they are connected. The Node View interface contains device types and device status selections as shown in Figure 14-4 on page 470.

Figure 14-4 Selecting a Node View

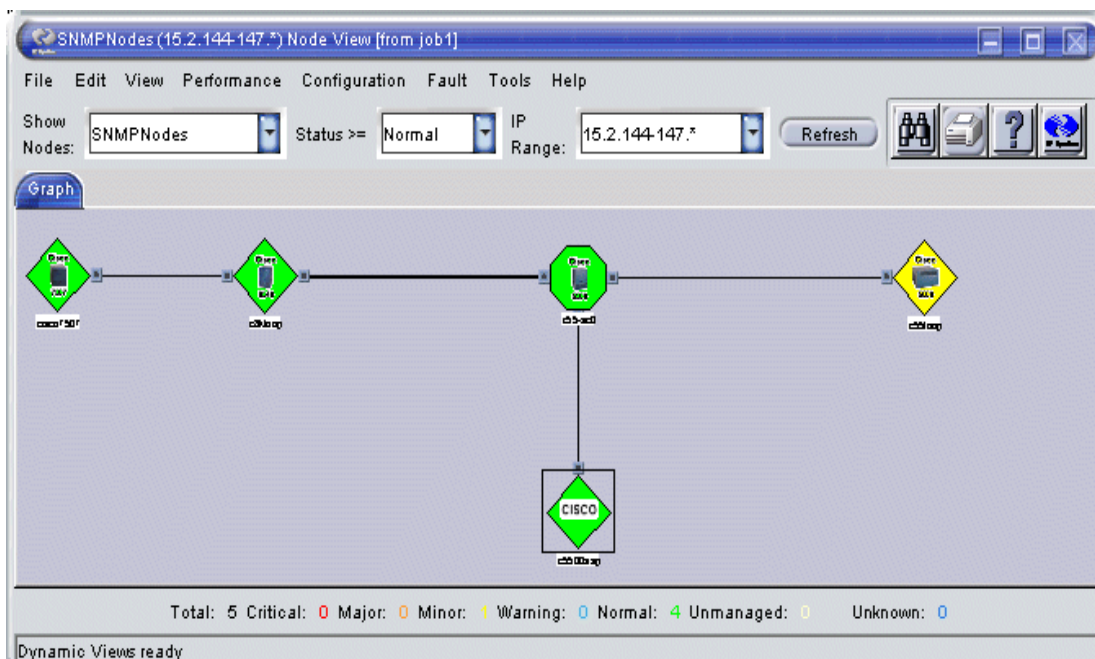


Device selection relies on filters contained within the `filters` file. This is the same file that contains filters such as discovery filters and DHCP filters. The filters file is found in the following locations:

- Windows: `install_dir\conf\C\filters`
- UNIX: `$OV_CONF/C/filters`

For detailed information about how to develop a filter, refer to *A Guide to Scalability and Distribution*.

**Figure 14-5** Node View Example



When working with Node view, placing your mouse over a node displays additional information about the node, such as the hostname, IP address, model number and version, system location, and system contact, subject to availability. Single-clicking allows you to zoom in and zoom out on the display. Double-clicking a node shows the node's attributes from the `ovtopodump` command.

### Station View

The Station view shows you a graphical representation of the NNM collection stations and NNM Advanced Edition management stations discovered in your topology. This view is useful only in a distributed management environment. See *A Guide to Scalability and Distribution* for information about distributed environments.

## Internet View

The Internet view shows you a graphical representation of the networks in your topology. With the Internet view, you can view the general status of your network and locate problems in your network.

## Port-Address Mapping

By selecting a node and using the `Tools:Port-Address Mapping` menu item you'll see a table of all devices connected to the selected node. The menu item works for most devices that support the *Transparent Forwarding Database MIB* or the *Bridge MIB*.

In Figure 14-6 the user selected device `4kfct5ue6m01` from the NNM map, then used the `Tools:Port-Address Mapping` command to display a table of all of the devices connected to the selected node along with information associated with these connected devices.

**Figure 14-6** Example of Port-Address Mapping Table

Port-Address Mapping for 4kfct5ue6m01.cnd.hp.com

Enter a node and click Port-Address Mapping to list information about devices heard by a connector device.

This is the Port-Address map for devices that are heard by [4kfct5ue6m01.cnd.hp.com](#).

Port	ifIndex	Hostname	Physical Address	Node Status
0	0	<a href="#">4kfct5ue6m01.cnd.hp.com</a>	0x0030C1448880	Normal
1	1	<a href="#">nsmd-gw1.cnd.hp.com</a>	0x00000C07AC00	Normal
1	1	<a href="#">Cisco-07AC02</a>	0x00000C07AC02	Normal
1	1	<a href="#">Cisco-07AC03</a>	0x00000C07AC03	Normal
1	1	Unknown	0x0030963011B7	Unknown
1	1	<a href="#">Cisco-F00934</a>	0x00B0C2F00934	Normal
1	1	<a href="#">fcs-5gwi.cnd.hp.com</a>	0x00B0C2F0096C	Normal
2	2	<a href="#">Cisco-C448CF</a>	0x003019C448CF	Normal
2	2	<a href="#">fcs-6gwi.cnd.hp.com</a>	0x00D0BCF6761C	Normal
3	3	<a href="#">2kswitch.cnd.hp.com</a>	0x0060B022687F	Normal
3	3	<a href="#">c8kloop.fc.hp.com</a>	0x00D0BA25CF17	Normal



## Modifying Dynamic View Menus

Application developers and end-users can modify Dynamic View menus. See the `menusettings.xml` (4) reference page in NNM's online help (or the UNIX manpage) for more information.

## Overview of the Java-based Web Interface

HP OpenView's Java-based web interface further expands NNM's network management capabilities. A part of NNM's base product, the web interface and its applications present the same information about your network that is available from the NNM management station. This information is now presented in a familiar, operator-friendly format that you can access from a remote workstation running a web browser. From your web browser on any PC or UNIX operating system workstation, you can log onto the web and access NNM on the management station to display maps and alarm information. You can easily monitor the status of your network from any place at any time, a real advantage when your pager goes off in the middle of the night!

The HP OpenView Web lets you:

- Inquire about the status of your network and troubleshoot problems from anywhere via the World Wide Web.
- Obtain dynamic updates on map information.
- Share management data with others.
- Present both a graphical and tabular view of the network.
- Access and configure security for web applications.
- Launch NNM and other web-based applications.
- Interact with a familiar web interface.
- Configure and view reports.

The HP OpenView Web includes the following applications:

- HP OpenView Launcher
- HP OpenView Network Presenter
- HP OpenView SNMP Data Presenter
- HP OpenView Alarm Browser
- HP OpenView ECS Configuration
- HP OpenView Correlation Composer
- HP OpenView Reporting interface

An overview and the administration of each of these applications is discussed in the following sections of this chapter.

## Setting Up the HP OpenView Web

When you install NNM on the management station the files and applications necessary to run the web interface are also installed. The management station must also have a web server running.

When you install NNM on a management station running Windows, a web server is not installed, since the Windows operating system comes with its own web server. The web server on the Windows operating system is automatically configured for NNM's use during the NNM installation.

When you install NNM on a management station with a UNIX operating system, a web server *is* installed. This server is fully configured to access scripting aliases and documentation aliases used by NNM. The web server uses port 3443 for communication.

If you want to set up your own web server, configure the port number and set up the aliases `OvCgi`, `OvDocs`, and `OvBackgrounds`, see NNM's release notes for additional information about installing a web server.

The client station only needs to have a web browser running. Supported web browsers are:

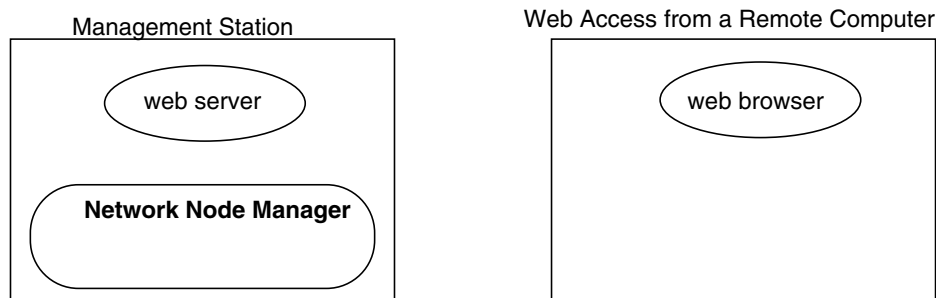
- Netscape
- Microsoft Internet Explorer

---

**NOTE** Refer to the "Supported Configurations: Supported Web Browsers" section of the NNM Release Notes for specific information about supported web browsers.

---

**Figure 14-7 Management Station and Remote Web Access**



See Figure A-4, "Services and Files for the Web Components," on page 535 for more detailed information about the files and services (processes) on the management station and remote computer at installation time.

On the management station, there must be a non-web based version of NNM running for each map that will be accessible through remote web access. This is shown in Figure 14-7. For example, if you want your team to be able to access the Home Office map and the map showing your operations in Germany, you must start NNM on the management station with your Home Office map displayed and start another instance of NNM on the management station with the Germany map displayed. See page 498 for more information.

From remote sites you can now use your web browser and NNM's Network Presenter to display the Home Office map or the Germany map.

### **Role Configuration Files for the HP OpenView Web**

Since the HP OpenView Web can be accessed from virtually any system, security measures are in place to ensure that only authorized users log in to your NNM management station. The way the HP OpenView Launcher and the web applications use these mechanisms varies. The security mechanisms are:

- User authentication (password) file.
- User authorization (roles) file.
- Session configuration file.
- Audit log files.
- `ovw.auth` and `ovwdb.auth` files (used by the Network Presenter and discussed in the section “Configuring Security for the Network Presenter” on page 496).

Since the Launcher is the access point for all other web applications, it provides the first level of security. When security is enabled via the session configuration file, the user password file and user roles file are read to ensure that the user is authorized to access NNM on the management station. The user roles file ensures that only those applications approved for that user role are made available via the Launcher.

### User Authentication (Password) File

User authentication is accomplished through a user name and user password. Users should not use the same password as their operating system password. Since the passwords are passed in cleartext format at the beginning of a session, it is possible that they may be intercepted. The Launcher uses the user password file when the user attempts to log on.

The user password file is located on the management station at:

Windows: `install_dir\www\etc\htpasswd`

UNIX: `/etc/opt/OV/share/www/etc/htpasswd`

Entries in the file appear as:

`john: FXSD198sdfADS`

`sue: PO12ADFpoiUS`

The file is installed with administrator-only (Windows operating system) or root-only (UNIX operating system) permission.

Set passwords as follows:

`ovhtpasswd username`

The command prompts you for the password for that user.

See the *ovhtpasswd* reference page in NNM's online help (or UNIX manpage) for more information about this command.

### User Authorization File

Users may be assigned to any number of groups for authorization purposes. These user groups are called user roles in NNM. A **user role** is a designation given to individuals who perform specific tasks with NNM. Users are assigned roles, and then those roles are given access to specific URLs. The Launcher uses the user roles file to define what list items are visible from the Launcher window. When a user logs in, his or her role is verified, and then he or she can access the URLs assigned to that specific role. User roles are valid only when a user accesses NNM via the web interface; they are not valid when a user accesses NNM directly on a management station.

Predefined user roles and their functions are listed below. You can change these roles, or add new roles.

- NetworkAdmin

The role of NetworkAdmin is intended for individuals who have a higher level of knowledge of networks. Capabilities for configuration of the network and connecting devices, and advanced troubleshooting should be associated with this user role.

- NetworkOper

The role of NetworkOper is intended for individuals who do routine troubleshooting and maintenance tasks associated with the network. Capabilities for monitoring the network, and routine troubleshooting and maintenance of the network and network devices should be associated with this user role.

- NTAdmin

The role of NTAdmin is intended for individuals who have a higher level of knowledge of the Windows operating system. Capabilities for configuration of the Windows operating system and advanced troubleshooting should be associated with this role.

- NTOper

The role of NTOper is intended for individuals who do routine troubleshooting and maintenance tasks associated with Windows operating system. Capabilities for monitoring the system, and routine troubleshooting and maintenance of the Windows operating system should be associated with this user role.

- UNIXAdmin

The role of UNIXAdmin is intended for individuals who have a higher level of knowledge of UNIX operating systems. Capabilities for configuration of UNIX operating systems and advanced troubleshooting should be associated with this user role.

- UNIXOper

The role of UNIXOper is intended for individuals who do routine troubleshooting and maintenance tasks associated with UNIX operating systems. Capabilities for monitoring the system, and routine troubleshooting and maintenance of UNIX operating system should be associated with this user role.

- OVAdmin

The role of OVAdmin is intended for individuals who do configuration and customization of the HP OpenView environment and management applications. Capabilities for configuring and customizing management applications should be associated with this role.

The user roles file is located on the management station at:

Windows: *install\_dir*\www\etc\htgroup

UNIX: /etc/opt/OV/share/www/etc/htgroup

The format of the user roles file is:

NetworkAdmin: user1 user2

NetworkOper: user4 user5

NTAdmin:user1 user3

UNIXOper: user6

You can also use a plus sign (+) instead of specific user names. This allows any user access to the capabilities of that user role.

User roles work in conjunction with the Access specification in the Launcher Registration Files (WLRG). See “Action Block” on page 486.

## Session Configuration File

The session configuration file is used by the Launcher. A **session** is a group of applications associated with a particular user on a particular display. When a user logs in, all the applications started (directly or indirectly) from the Launcher are part of that session. The applications that are part of a session can share information, such as locale.

A session is terminated when the user exits the web browser, when the session has been inactive for some configurable period of time (the default is 9 hours), or when the management station is shut down.

The session configuration file allows you to enable or disable security for NNM's web applications (default is disabled). If you specify `Off` in the `UserLogin` field of this file, NNM's web applications open directly. If you specify `On` in the `UserLogin` field of this file, NNM's web applications display a login dialog box upon opening.

---

### WARNING

**The default is for the session configuration file to disable security. If the security mechanisms are disabled in this file, users will not be presented with a login screen, and will have access to all registered actions.**

---

The session configuration file is located on the management station at:

Windows: `install_dir\www\conf\session.conf`

UNIX: `/etc/opt/OV/share/www/conf/session.conf`

The session configuration file has the following format:

UserLogin: `on|off`

LoginLogging: `on|off`

AccessLogging: `on|off`

SessionTimeout: `hours`

The default values are `off` for the first three items, and 9 hours for the session timeout. Time should be specified in integer values. Open the editable ASCII `session.conf` file for more information.

If you modify this file while you have the Launcher running, you need to exit the web browser and restart the Launcher for the new information to take effect.



## Audit Log Files

There are two audit log files — one for logging logins and one for logging URL access from the Launcher. These are useful for security auditing purposes, as you can check on which users are accessing the Launcher by monitoring these files.

There is no log file management; the files continue to grow. You should periodically check these files and when they get large, you should save them or trim them.

The files are located on the management station at:

Windows: *install\_dir*\www\logs\login\_log (login log)

*install\_dir*\www\logs\access\_log (access log)

UNIX: /var/opt/OV/www/logs/launch/login\_log (login log)

/var/opt/OV/www/logs/launcher/access\_log (access log)

Each log entry is a single line of text. If the session configuration file is set to LoginLogging:on, the following information is recorded:

- Host
- User name
- Date
- Session number
- Access permitted; either Allowed or DENIED
- URL accessed (for the access\_log file only)

## The HP OpenView Launcher

The HP OpenView Launcher is the user interface integration point for the HP OpenView web applications. Just as with management station-based NNM, developers can tie their applications into NNM's web interface. Then users access those applications through the Launcher interface.

You can also create your own application that you access from the Launcher. If you create your own application, you can integrate it with the Launcher by creating a registration file for that application.

The Launcher provides functionality in three areas:

- Launching of management functionality and applications via a URL or via the web interface.
- User login and user roles.
- Maintenance of session information shared by all web interfaces.

### Launcher User Interface

The Launcher interface is started from a web browser. The URL for the Launcher is:

```
http://hostname:[port]/OvCgi/ovlaunch.exe
```

For example, if the host system is a management station named `UX-manage1` running a UNIX operating system, you would enter:

```
http://ux-manage1:3443/OvCgi/ovlaunch.exe
```

A web server is automatically installed on UNIX management stations and configured to use port number 3443 for communication.

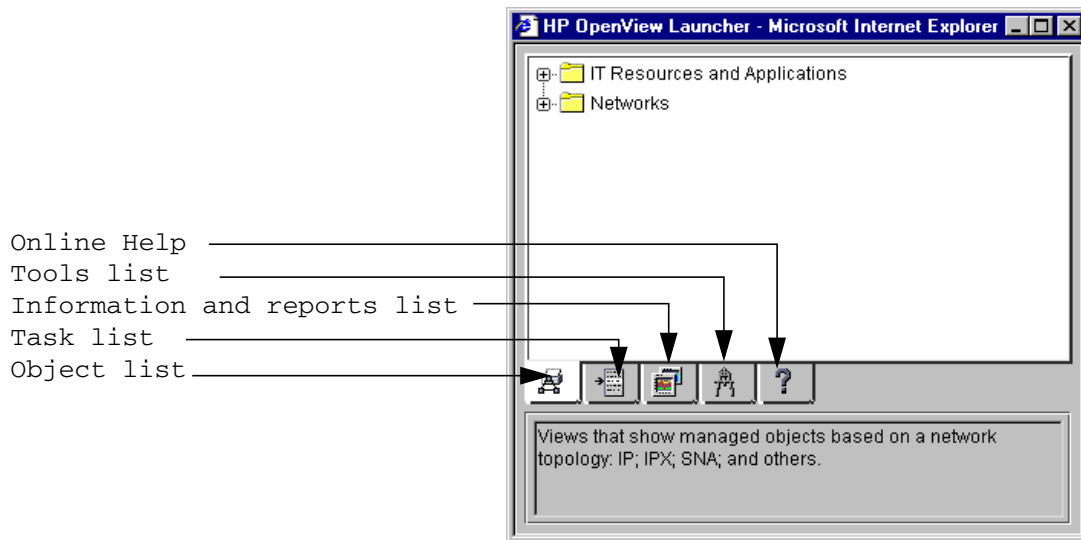
If the host system is a workstation named `Windows-mgmt` running Windows, you would enter:

```
http://Windows-mgmt/OvCgi/ovlaunch.exe
```

No port number is required for management stations running a Windows operating system.

If the management station has been configured for security, a login screen appears where the user enters his or her name and password. If no security has been configured, the Launcher window comes up without asking for login and password.

**Figure 14-8**      **The Launcher Window**



The Launcher window displays a tabular view of the capabilities available through the Launcher. The bottom tabs of the Launcher window contain the Task, Tool, Object Views, Management Areas, and Help categories. These categories correspond to the basic functionality of NNM on a management station. These are also the categories where other HP OpenView web applications will be linked.

Click on a category to expand it, or click on the expanded category to contract it. You can also click on the +/- indicator.

If you double-click on a list item, you execute the URL associated with that item.

The categories that are displayed in the tabular section of the Launcher window are determined by the user roles file and the Launcher Registration Files (WLRF), described next.

## Configuring the Launcher

There are several files you can use to customize the Launcher.

- Launcher registration files.
- Error log file.
- Installed locale file.
- Java parameters file.

### Launcher Registration File

The Launcher registration files are used to configure aspects of the appearance and behavior of the Launcher interface.

The format of the Web Launcher Registration Files (WLRF) closely follows the syntax and constructs of the HP OpenView Application Registration File (ARF). Refer to the online manual *Creating and Using Registration Files* for general information about Application Registration files. Refer to the *LauncherRegIntro* reference page in NNM's online help (or the UNIX manpage) for specific information about creating or editing the WLRF files.

The WLRF files are located on the management station at:

Windows: `install_dir\www\registration\launcher\C\*.*`

UNIX: `/etc/opt/OV/share/www/registration/launcher/$LANG`

The bitmaps referenced in the WLRF are located in:

Windows: `install_dir\www\htdocs\C\images\*.*`

UNIX: `/opt/OV/share/www/htdocs/$LANG/bitmaps`

**Components of the WLRF** There are several sections of a WRLF that you might need to modify. Only those sections will be discussed here.

Once you have modified a WLRF file, you can check it for errors with the command `regverify`. See the *regverify* reference page in NNM's online help (or the UNIX manpage) for more information. This command is located on the management station at:

Windows: `install_dir\bin\regverify`

UNIX: `/opt/OV/bin/regverify`

## Tab Block

The Launcher has several tabs at the bottom of its window. The tab block indicates under which of the tabs an application will appear.

The tab block allows an optional icon file name to display on the tab. If one is not specified, a default will be used. The icon appears in front of the name of the application in the Launcher content area. The format for icons is 16x16 GIF. The file name is relative to:

Windows: *install\_dir*\www\htdocs\bitmaps

UNIX: *\$OV\_www*/htdocs/bitmaps

The tab block also contains optional active help text. This active help is presented when the mouse pointer is over the tab.

The tab block contains List Item entries instead of Menu Items (for the menu bar). Following is a sample of the tab block.

```
Tab <60> "Tools" Icon "launcher/toolsfold.20.gif"
      ActiveHelp { "tools" }
{
  <70> "Session"
      Icon "launcher/toolsfold.16.gif"
      ActiveHelp {"Session Info" }
      f.list "SESSION";
}
```

## List Block

The list block of the WLRF corresponds to the menu block in the ARF.

The following statements are allowed in the list block:

- Precedence value  
This has the same definition as in the ARF.
- List Item Label  
This is the label that is displayed for the entry.
- Icon  
This is a locale-dependent icon that is displayed for the list item. The format for icons is 16x16 GIF.
- ActiveHelp  
This is the active help text that is displayed when the mouse pointer is placed over the entry.

- Functions

Two functions are supported:

— `f.action`

This indicates a terminal list item. It points to an Action block that has the action definition.

— `f.list`

This indicates a component list item. It points to a list block to allow the definition of a hierarchical tree list.

Following is a sample of the list block.

```
List "SESSION"{
    <80> "Session Info Viewer"
        Icon "launcher/tools.16.gif"
        ActiveHelp { "Session Info Viewer" }
        f.action "printsession";
}
```

### Action Block

The action block of the WLRP serves the same function as in the ARF. The following statements are allowed in the action block:

- URL

This is the URL to launch for the action. The URL may contain the special variable `$OVWebServer`. This variable will be substituted with the name of the system from which the Launcher was loaded.

- Access

This is a list of user roles that have access to this action. If a user does not belong to a group that has access, the user will not see the list item that invokes this action in their Launcher window. If the access statement is not present, all valid users can access this action. The user roles values correspond to entries in the user roles file. For example, `NetworkAdmin`. See “User Authorization File” on page 478.

- Window

The Window statement specifies the characteristics of the window into which the URL will be loaded. If the `WebWindow` statement is not present, the URL will be loaded into an unnamed full web browser window.

The Window definition has the following components:

— Window name

This is a name for the window to allow window reuse by targeting a named window. The window name is a string of alphanumeric characters and underscores; it may not include blanks. If a new window is desired every time the URL is loaded, a blank window name (" ") may be given.

— Type (full, intermediate, or limited)

Type specifies the type of the window and determines which features it supports. If no type field is present in a window block, full is assumed. The values are:

— full, all window elements present and user configurable.

— intermediate, directories and menubar off, all other window elements present and user configurable.

— limited, all window elements off but user configurable.

— Toolbar (on or off)

— Location (on or off)

— Status (on or off)

— Scrollbars (on or off)

— Resizable (on or off)

— Width (integer value)

— Height (integer value)

Following is a sample of the action block.

```
Action printsession
{
    URL "/OvCgi/printsession.exe";
    WebWindow "OvPrintSession" {
        Type full;
        Toolbar off;
        Status off;
    }
}
```

## Error Log File

The Launcher error log records internal errors from the Launcher. You can use the output from this file when discussing a problem with the HP Response Center or HP support personnel.

The Launcher error log file is located in:

Windows: *install\_dir\www\logs\launcher\error\_log*

UNIX: *\$OV\_WWW/launcher/error\_log*

The error log contains the following information:

- Date
- Host
- User
- Session
- Reason or error

The error log file continues to grow without bounds. You need to watch the size of this file and truncate it as needed.

If the error log file does not give you sufficient information for troubleshooting, you can set a parameter in the URL for **ovlaunch.exe** that will enable more in-depth error logging on a session level. See “Troubleshooting Web Components” on page 576 for more information. The parameter is:

**http://sysname:3443/OvCgi/ovlaunch.exe?Debug=/tmp/file**

## Language Selection

The Launcher and the Web applications started from the Launcher execute under a particular language, such as English or Japanese. The language used by the Launcher and the Web applications is based on the following:

- The language specified by the user when the security is enabled.
- The language specified by the user through the URL.
- The language specified by the browser.

Once a session is started, the language selection cannot be changed without restarting the browser with a different language.



**Language Selection When Security is Enabled** Security is enabled via the session configuration file. When security is enabled, the user is presented with a login page listing the languages available for the session under the [Options] button. This list represents the languages supported by the Launcher and other installed products that integrate into the Launcher. The languages presented in [Options] is based on the contents of the `locales.installed` file. This file is located in:

Windows: `install_dir\www\conf\locales.installed`

UNIX: `/etc/opt/OV/share/www/conf/locales.installed`

This file contains a list of locales for which the Web applications have been localized. For instance, if the Japanese NNM has been installed, this file will contain an entry for English and for Japanese, for example:

en

ja

This file can be modified by a developer or administrator. When an HP OpenView web application has been localized to a language not listed in this file, a locale entry should be added to this file. Refer to the *HP OpenView Windows Developer's Guide* for details on syntax.

**Language Specified Through the URL** The language for a session can be specified when the `ovlaunch.exe` URL is specified. This is done through the CGI parameter `AcceptLang`. This language specification will take precedence over the browser language specification, but can be overridden if security is on and the user chooses a different language on the login page. If security is not on, the login page will come up in the language specified by the `AcceptLang` parameter. For example:

`http://mysystem:3443/OvCgi/ovlaunch.exe?AcceptLang=ja`

**Language Specified Through the Browser** When security is not enabled, the Launcher uses language specified through the browser. For instance, if the browser has been configured to run in English, the Launcher will execute in English.

### Java Parameters File

There are some Java parameters that you can modify to control Launcher appearance. You change these parameters by modifying the `APPLET` block in the following file, located on the management station at:

**Windows:** `install_dir\www\htdocs\C\nm\launcher\browser.html`

**UNIX:** `$OV_www/htdocs/$LANG/nm/launcher/browser.html`

The Java parameters that you can modify are:

- background
- foreground
- activeTabColor
- inactiveTabColor
- treeBackground
- treeForeground
- treeConnectingLineColor
- displayActiveHelp
- activeHelpLines
- fontName
- fontStyle
- fontSize
- displayInFrame

---

## HP OpenView Network Presenter

The HP OpenView Network Presenter is a Java-based applet that connects to NNM for the purpose of displaying maps. It provides multiple views of map data on a management station from any system with a web browser that supports Internet Explorer or Netscape Navigator.

---

### NOTE

Refer to the "Supported Configurations: Supported Web Browsers" section of the NNM Release Notes for specific information about supported web browsers.

---

The Network Presenter communicates with the management station to get registration information, map data, and map object data. It provides read-only access to the map data and has dynamic updates to reflect the changes that occur on the management station. You can navigate through the map views, select and obtain status on symbols, and launch other web applications on the selected objects via the Network Presenter.

### Starting the Network Presenter

You must start an NNM session on the management station for each map that you want available through the web, as described on page 475.

You can invoke the Network Presenter through the Launcher using either of the following tabs:

Objects tab, then choose Networks:IP Network

Tools tab, then choose NNM:Network Presenter

Alternatively, you can enter the URL for the Network Presenter:

```
http://hostname[:port]/OvCgi/jovw.exe?[MapName=mapname&ObjectName=selectionname]
```

On UNIX operating systems, you must specify port number 3443. On a Windows operating system, no port number is required. The MapName and ObjectName fields are optional.

The MapName field specifies which map to open. The ObjectName field specifies what object to show initially. Refer to the *OvWebURLIntro* reference page in NNM's online help (or UNIX manpage) for more information.

---

**NOTE**

When starting the Network Presenter directly from the URL, you lose the session information that the Network Presenter would have if it were started from the Launcher, including language information.

---

If a *mapname* is not specified in the URL used to start the Launcher, Network Presenter attempts to connect to an NNM session running with the default map. If there is no NNM session running with the default map, the Network Presenter displays an error message that says the map was not found.

If a *mapname* is specified in the URL, the Network Presenter attempts to connect to an NNM session on the management station with an open map called *mapname*. If there is no NNM session running on the management station with this map open, the Network Presenter displays an error message that says the map was not found.

If you close the NNM session or open a new map, the Network Presenter session attached to that NNM session is ended. The Network Presenter displays a dialog box informing the users that NNM is closing its map.

By default, the Launcher will start the Network Presenter with the default map displayed. To make a map other than default available to the Network Presenter, modify the `jobw` Web Launcher Registration File (WLRFile) that specifies in the URL portion which map to open. See “Listing Multiple Maps in the Launcher” on page 498 and “Launcher Registration File” on page 484 for information.

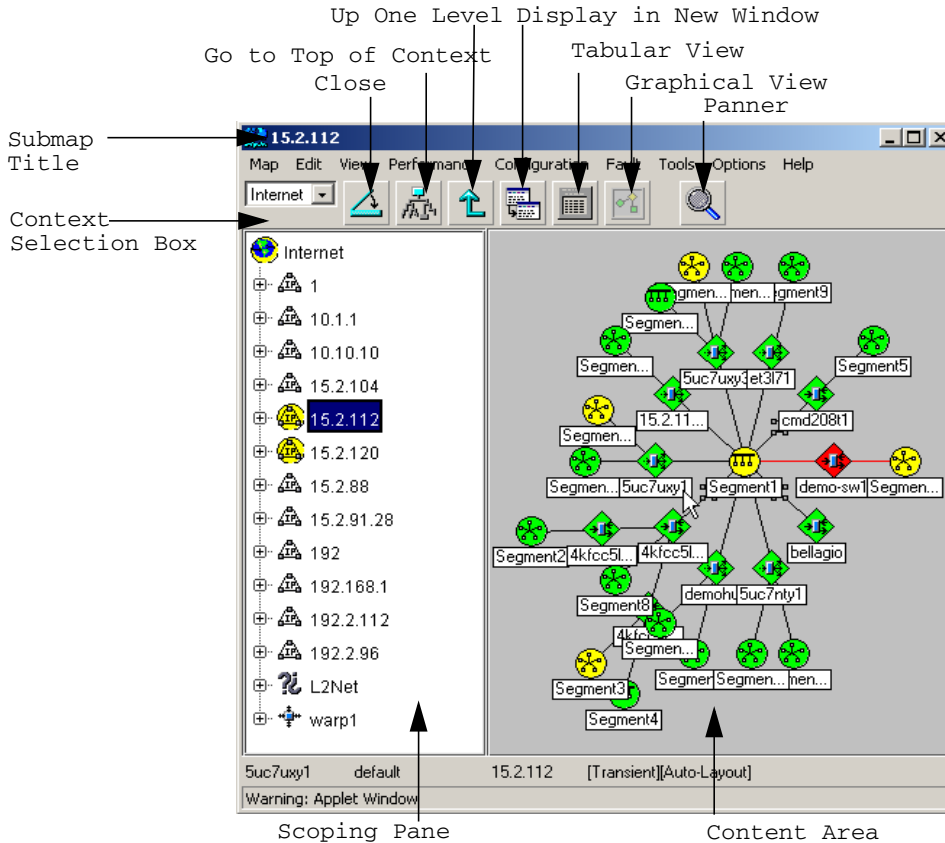
## The Network Presenter Window

The Network Presenter window has the following components:

- Submap title
- Menu bar
- Context selection box
- Toolbar
- Content area
- Scoping pane

- Status bar

**Figure 14-9 Network Presenter Window**



## Network Presenter versus NNM on a Management Station

There are some differences in the functionality and presentation of information between the Network Presenter and NNM on a management station.

### Features in the Network Presenter

Following are features in the Network Presenter that are not available in NNM on a management station.

- Context selection box to choose among the highest-level views.
- Scoping pane.
- Tabular view of data.
- Scroll bar in content area.
- New toolbar buttons: [Display in New Window], [Tabular], [Graphical]
- New menu items: Up One Level, Top of Hierarchy

**Features That Differ in the Network Presenter**

Some features from NNM on a management station are presented in a different format in the Network Presenter.

**Table 14-1 Differences in Features Between NNM and Network Presenter**

NNM Management Station Feature	Network Presenter Feature
[Go to Root] toolbar button	[Go to Top of Context] toolbar button
Parent toolbar	Use [Up One Level] button
App-defaults file	Java properties file
Symbol labels	Truncated
Background graphics	GIF or JPEG only
Inner shapes	GIF only
Open map	Start another Network Presenter session
Persistent and transient submaps	Supported; cannot change transient to persistent
Automatic layout	Cannot turn on or off

**Features Not Available in the Network Presenter**

The following features in NNM on a management station that are not available in the Network Presenter.

- Map Description dialog box.
- Submap Properties dialog box.
- Locate Submap dialog box.
- Home submap.
- Symbol Properties dialog box.
- Snapshots.

- Symbol enhancements (flashing, transparent, alerts, etc.).
- Menu collapsing.
- Symbol, submap editing.
- Invoking executable symbols.

### **Interaction Between NNM on the Management Station and the Network Presenter**

Since the Network Presenter cannot do editing, the Network Presenter does not influence what happens to NNM on the management station; however, changes in the current session on the management station are dynamically reflected in the content area and scoping pane of the Network Presenter as appropriate. You will see changes in the content area if the submap that is displayed on the management station changes. You will see changes in the scoping pane if that level of hierarchy is displayed. Changes are not dynamically reflected in open dialog boxes.

### **Configuring the Network Presenter**

You can modify the Network Presenter Registration Files (NPRF) to configure certain aspects of the Network Presenter.

The NPRFs are located on the management station at:

Windows: `install_dir\www\registration\jovw\c\*.*`

UNIX: `/etc/opt/OV/share/www/registration/jovw/$LANG`

The format of the Network Presenter Registration Files (NPRF) closely follows the syntax and constructs of the HP OpenView Application Registration File (ARF). Refer to the online manual *Creating and Using Registration Files* for general information about Application Registration files. Refer to the *NetworkPresenterRegIntro* reference page in NNM's online help (or the UNIX manpage) for specific information about creating or editing the NPRF files.

You can add entries to the Action block to specify your own applications to be called when the Network Presenter is launched. You can also change the WebWindow properties in the Action block to control some aspects of the appearance of the Network Presenter (see "Components of the WLRF" on page 484 for sample contents of a registration file). You can add pull-down menus, toolbar buttons, and popup menus that operate on selected objects.

## Configuring Security for the Network Presenter

There are two levels of security for the Network Presenter.

- The `session.conf` file, described earlier in this chapter.
- The `ovw.auth` and `ovwdb.auth` files, described next.

The Network Presenter allows role-based, restricted access to menu items that have action blocks that contain access statements. See the *Action Block* description on page 486 for more information.

**Authorization Files** The `ovw.auth` and `ovwdb.auth` files specify which users are allowed access to which management stations. These files are located on the management station at:

Windows: `install_dir\conf\ovw.auth`

`install_dir\conf\ovwdb.auth`

UNIX: `$OV_CONF/ovw.auth`

`$OV_CONF/ovwdb.auth`

Each of these files contain instructions for their use.

## Using Symbol Registration Files and Bitmaps

Symbol registration files used by the Network Presenter are the same as those used by NNM on a management station. Therefore, if a symbol definition is added for NNM on a management station, it also appears in the Network Presenter. Symbol registration files are located in:

Windows: `install_dir\symbols\%LANG%`

UNIX: `$OV_SYMBOLS/$LANG`

See the *OVwRegIntro* reference page in NNM's online help (or the UNIX `manpage`) for more information about creating or editing symbol registration files.

The graphic files for the symbols, themselves, are stored in two places. One directory used by NNM on the management station (see page 626), and one directory used by NNM's web-based interface. The web-based interface requires GIF files. Other additional graphic file formats are supported on the management station. If you create your own GIF files, place a copy in each location.



The Network Presenter loads its bitmaps in GIF format from the directory:

Windows: *install\_dir*\www\htdocs\bitmaps

UNIX: *\$OV\_WWW*/bitmaps

---

**TIP**

If you can see a symbol on the management station, but cannot see it over the web interface, check the `Contrib` directory (see page 56) for the `convertBitmaps.ovpl` script which walks through each graphic in the management station's symbol graphic directory and ensures that a corresponding GIF file exists or is created in the web interface's symbol directory.

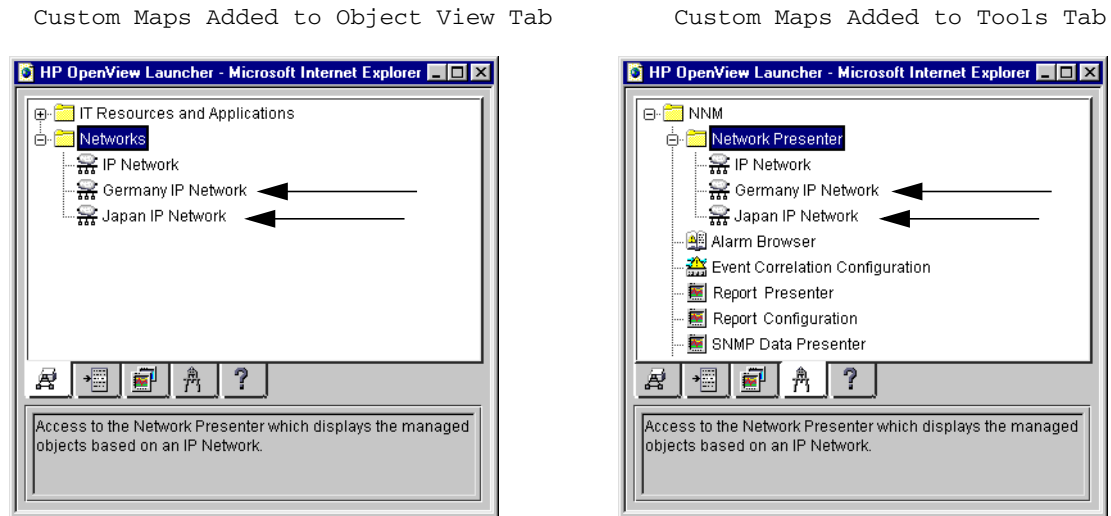
---

Symbols in the web-based interface have the same appearance as in NNM on a management station; however, the symbol labels do not scale based upon the size of the label. Instead, a long label is truncated to fit the space available.

## Listing Multiple Maps in the Launcher

You can modify the Launcher registration file for the Network Presenter so that each of your custom maps is listed.

**Figure 14-10** Listing Multiple Maps in the Launcher Window



For each map, you must make additions to the List and Action blocks of the following file:

- Windows: `install_dir\www\registration\launcher\C\joww`
- UNIX: `$OV_WWW_REG/launcher/$LANG/joww`

Open this file and read the instructions provided for more information. See also the *LauncherRegIntro* reference page in NNM's online help (or the UNIX manpage) for specific information about creating or editing the WLRf files.

On the management station, there must be a non-web based version of NNM running for each map that will be accessible from the Network Presenter. For example, if you want your team to be able to access the Germany map and the map showing your operations in Japan, you must start NNM on the management station with your Germany map displayed and start another instance of NNM on the management station with the Japan map displayed.

## SNMP Data Presenter

The SNMP Data Presenter displays results of operations that present tabular or textual data for managed SNMP nodes selected from the map.

You can use the SNMP Data Presenter even if there is no NNM session running on the management station. The SNMP Data Presenter accesses NNM's Application Registration Files (ARFs) to retrieve information about menu structure and actions, executes the command on the management station, and displays the results in the web browser.

The SNMP Data Presenter displays configuration, performance and fault information. Here are a few examples:

- Interface status.
- SNMP authentication failures.
- TCP connections.
- IP Network addresses.
- IP Routing tables.
- IP ARP cache table.
- System information.
- SNMP trap recipients.

To start the SNMP Data Presenter, select the target node in the content area of Network Presenter and then select one of the following:

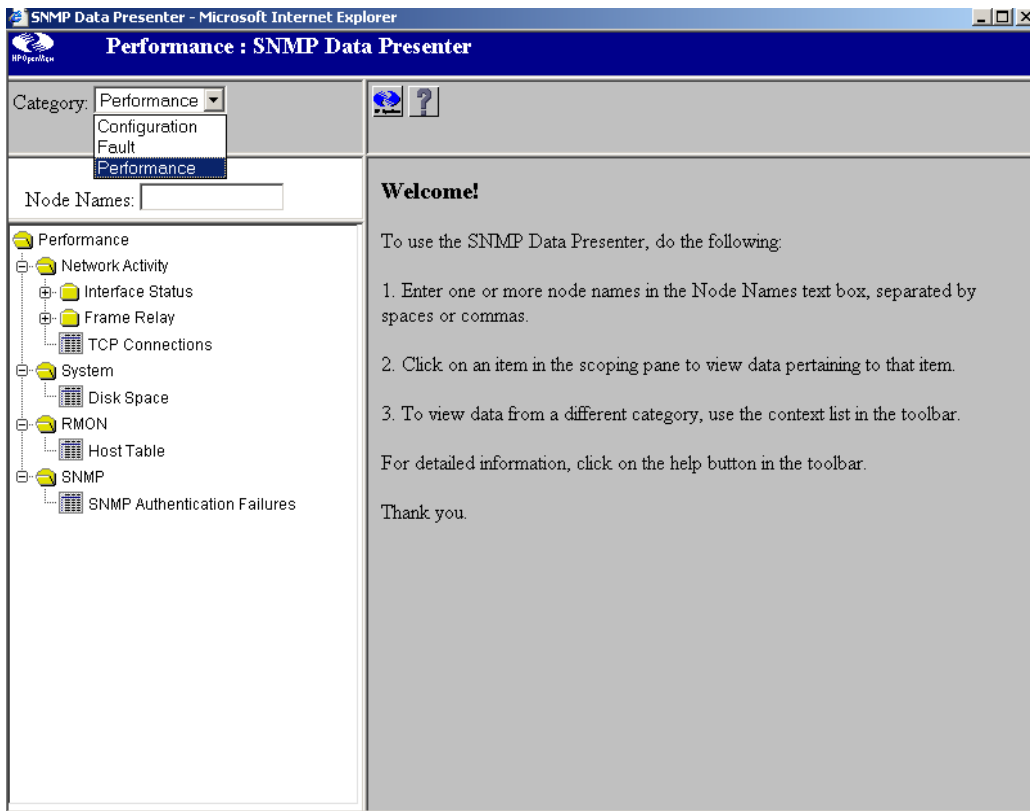
- Performance:SNMP Data Presenter
- Configuration:SNMP Data Presenter
- Fault:SNMP Data Presenter

To start the SNMP Data Presenter from the Launcher, select:

- Tools tab, then NNM:SNMP Data Presenter

The SNMP Data Presenter window is shown in Figure 14-11.

**Figure 14-11** SNMP Data Presenter Window



Scoping Pane

Content Pane

When accessed from the `Tools` tab, the scoping pane contains categories and actions in a tree format. You can click on the category items to expand them and then click on an action to execute that action. You can also use the Category pull-down.

The results of an action are shown in the content area. A link at the bottom of this area allows you to display a new web browser window with the currently displayed information. This allows for easy comparing of data.

## Configuring the SNMP Data Presenter

The SNMP Data Presenter uses NNM's Application Registration Files (ARFs) and the Application Builder. Because of this, you can execute the same SNMP monitoring operations as you do in NNM, and add new operation commands to the menu structure with Application Builder.

See NNM's online manual, *Creating and Using Registration Files*, for details about editing application registration files (ARFs). See "Using the MIB Application Builder" on page 425 for more information about the Application Builder feature.

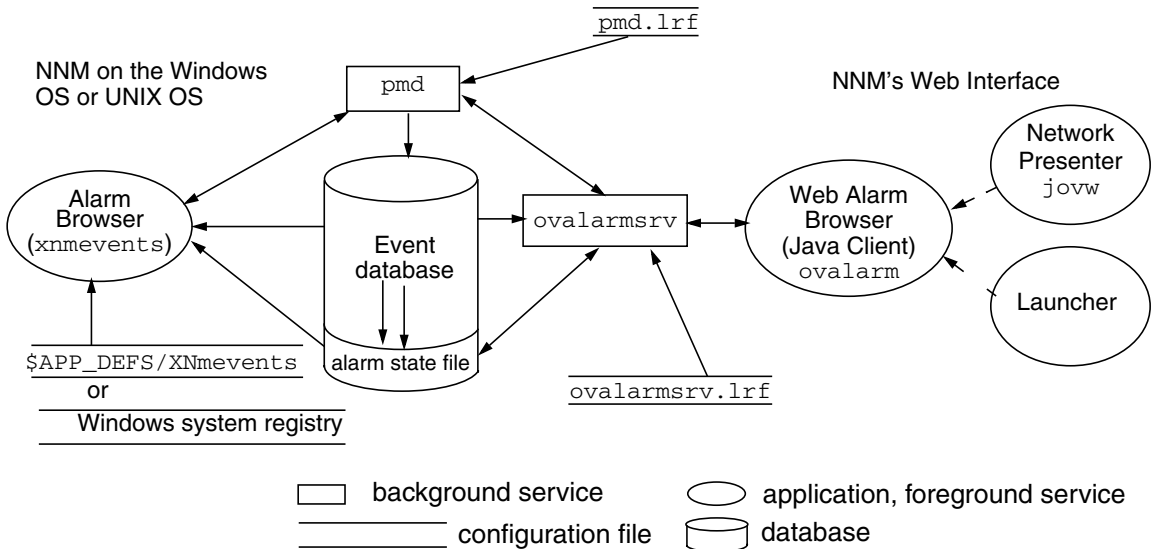
## Alarm Browser

The Alarm Browser presents alarm data via the web interface. Through the Alarm Browser, you can filter alarms and take actions on specific alarms. All alarms are global to all web users, as are the acknowledge, delete, change category, and change severity functions.

The web-based Alarm Browser service, `ovalarmsrv`, is started by `ovstart` when you start an NNM session on the management station.

Figure 14-12 illustrates the relationship and data flow between the Alarm Browser application on a management station and the web-based Alarm Browser.

**Figure 14-12 Relationship Between the Two Alarm Browsers**



You access the Alarm Browser using one of the following methods:

- In the Launcher, select the Tools tab, then select NNM:Alarm Browser

- In Network Presenter, select the **Fault** menu, then select **Alarm Browser**.

**Figure 14-13** Web-Based Alarm Browser's Category Window



## Using and Configuring the Alarm Browser

You can use the Alarm Browser to perform the following:

- Acknowledge/Unacknowledge alarms.
- Move alarms to a different category.
- Assign alarm severities.
- Filter alarms.
- Delete alarms.
- Examine alarm details to find correlated events.

To move an existing alarm message to a different category. Select the alarm that you want to reclassify. Then select **Actions:Change Category** from the menu bar. Refer to the online help if you need more information.

This change affects only the selected alarms. Future alarms of the same type will continue to be posted in the old category's list.

You can configure an alarm to activate your web browser to a specific URL. See "Launching Specific Views or URLs from Alarms" on page 327 for more information.

See Chapter 12, Customizing Events: Doing It Your Way, for configuration options:

- Setting the maximum size of the event database (default 16 MB).
- Setting the maximum number of alarms in the Alarm Browser's state file (default 3500).
- Setting the number of alarms that should be deleted each time the state file reaches maximum capacity.
- Copying or restoring the Alarm Browser's state file.
- Assigning alarms to other categories and creating new alarm categories.

### **Security for the Alarm Browser**

The Alarm Browser uses the user login security mechanisms, described earlier in this chapter.



---

## Event Reduction

Event reduction is the process by which the relationship between events is identified. Once identified, a smaller number of new events providing higher information content can be generated. This simplifies the task of diagnosing network faults. There are three event reduction strategies that are configured and working within NNM. You can access the configuration windows for two of them over the web:

- “ECS Correlations” on page 349
- “Correlation Composer Correlators” on page 371

There are several ways to access these event correlation features. For more information, from any submap:

- Select Tools:HP OpenView Launcher.
- Select the [?] tab.
- Click Tasks, Event Correlation Management.

Read the information under *Accessing the Event Correlation Configuration Windows*

For information about the NNM event reduction strategies, see Chapter 11, “Event Reduction Capabilities: Getting to the Root Cause,” on page 339.

## Reporting Interface

The web Reporting interface is a graphical user interface that is available through a web browser. The Reporting interface consists of:

- Report Configuration-- create, modify, and view report schedules.
- Report Presenter-- view reports.

You can access the Report Configuration in one of the following ways:

- From NNM on a management station, select the `Options:Report Configuration` menu item.
- From Network Presenter, select the `Options:Report Configuration` menu item.
- From the Launcher, select the `Tools` tab, and expand the NNM category. Double-click on `Report Configuration`.
- From your browser, specify the following URL:

```
http://hostname[:port]/OvCgi/RptConfig.exe
```

You need to specify a port if you are running NNM on a UNIX machine.

You can access the Report Presenter in one of the following ways:

- From NNM on a management station, select the `Tools:Report Presenter` menu item.
- From Network Presenter, select the `Tools:Report Presenter` menu item.
- From the Launcher, select the `Tools` tab, and expand the NNM category. Double-click on `Report Presenter`.
- From your browser, specify the following URL:

```
http://hostname[:port]/OvCgi/RptPresenter.exe
```

You need to specify a port if you are running NNM on a UNIX machine.

The NNM Analysis and Reporting interface will be displayed in your browser window.

Refer to the online manual *Reporting and Data Analysis with HP OpenView NNM* and the web online help for more information.



---

# **15**      **Maintaining NNM**

This chapter gives you ideas and suggestions for maintenance tasks that, when performed on a regular basis, can help keep Network Node Manager running more smoothly and without errors.

Included are tasks for:

- Daily maintenance, including checking running services, checking disk space, and trimming files (page 511).
- Weekly tasks, including backing up NNM, cleaning up databases, and monitoring polling performance (page 515).
- Monthly tasks, including obtaining the latest patch releases and updating licensing requirements (page 517).
- Yearly tasks, including evaluating the latest product release (page 518).
- Other maintenance tasks that will help your NNM installation run smoothly (page 519).

For information about maintaining NNM's data warehouse, see the online manual *Reporting and Data Analysis with HP OpenView Network Node Manager*.

---

## Daily Tasks

You may wish to perform the following tasks on a daily basis.

### Check Running Services

Each day you should check to see that all the required services (processes) are running. It is possible that a service has gotten “hung” or did not restart after a backup.

To check on the status of services use the `ovstatus` command. See the *ovstatus* reference page in NNM’s online help (or the UNIX manpage) for command options.

If you encounter a service that fails on a regular basis and you cannot find a cause, refer to the service’s reference page in NNM’s online help (or the UNIX manpage) for more information. You can also use the Event Configuration feature to automatically restart the service whenever it fails. In the Event Configuration dialog box, open the OpenView: `OV_DaemonTerminated(.1.3.6.1.4.1.11.2.17.1.59180100)` event configuration and make the following entry in the Command for Automatic Actions field: `ovstart $4`.

### Check Disk Space

You should check the available disk space on a daily basis. This will ensure that you have enough space to store collected data and to perform a backup.

To check available disk space:

- Use the Performance: System->Disk Space menu item.

Since this information is generated by an HP enterprise-specific MIB, it is available only from an HP 9000 or Sun SPARCstation running the HP OpenView SNMP Agent software.

If this menu item is not available, you can use the following:

*Windows:* Use Windows Disk Administrator

*UNIX:* Use the `df` or `bdf` command. See the appropriate manpage for more information.

To regain disk space, you can trim certain files, described next.

## Trim Files

Some files can grow large very quickly. This slows the discovery and map generation processes and affects overall system performance.

### The NNM Data Warehouse and snmpCollect

NNM pre-configures several data collections for you. NNM exports data from pre-configured data collections as well as collections you have manually configured, to the data warehouse. NNM uses data from the data warehouse to generate pre-configured General Availability, Threshold Violations, General Inventory, and Ping Response Time/Ping Retries reports. NNM automatically trims data from the following directories as well as from the data warehouse on a regular basis.

- *Windows:* `install_dir\databases\snmpCollect\`
- *UNIX:* `$OV_DB/snmpCollect/`

---

#### NOTE

Select Options: Data Collection and Thresholds to see the list of current data collections. Event information is also exported to the data warehouse and trimmed from the Binary Event Store regularly. NNM automatically trims both collected data and event information to reduce disk usage.

---

#### TIP

HP uses trimming options that will satisfy a majority of your requirements. If you need to keep collected data and/or Data Warehouse data for longer time periods, you may need to manually complete the export and trimming functions using the `ovdwtrend` command. See the bulleted items on page 510.

---

To reduce the rate at which the `snmpCollect` directory fills up all available disk space, do the following:



- Reduce the polling intervals by using the `Options:Data Collection & Thresholds` operation. This slows the rate at which `snmpCollect` fills up available disk space. In addition, if you want to check fixed thresholds only, do not store the data.
- You can use the `ovdwtrend` command to reduce the size of files residing in the `snmpCollect` directory. For example, you can use the `ovdwtrend -export -delpriorto "2001-12-31 00:00:00"` command to export all new data collected since the last export while trimming `snmpCollect` data file information dated prior to December 31, 2001. Review the *ovdwtrend* reference page in NNM's online help (or the UNIX manpage) to plan your strategy for trimming files in the `snmpCollect` directory.
- You can use the `ovdwtrend -trim` command to delete SNMP trend data from the NNM data warehouse. For example, you can use the `ovdwtrend -trimpriorto [240]` command to trim data older than 240 hours from the *snmp\_reduced\_trend* table of NNM's data warehouse. Review the *ovdwtrend* reference page in NNM's online help (or the UNIX manpage) and the online manual *Reporting and Data Analysis with HP OpenView Network Node Manager* to plan your strategy for managing data in NNM's data warehouse.

## Removing Reports

NNM automatically generates General Availability, Threshold Violations, General Inventory, and Ping Response Time/Ping Retries reports. You should review these reports and remove those reports that are no longer of use. See the online manual *Reporting and Data Analysis with HP OpenView Network Node Manager* for instructions on how to delete reports.

## Trace and Log Files

Trace and log files can grow quite large, and should be periodically checked for size.

Tracing, especially in the case of `netmon`, can cause the trace file to grow extremely large (multiple megabytes in size) fairly quickly. If you use tracing, remember to monitor the size of the trace file frequently, and turn tracing off as soon as you are finished. The `netmon` trace file is `install_dir\log\netmon.trace` (`$OV_LOG/netmon.trace`).

Windows: Run **netmon -MO** to disable polling. Use Notepad to edit the *netmon.trace* file, deleting all lines. Save the file. Then run **netmon -M** to enable polling.

UNIX: **cat /dev/null > \$OV\_LOG/netmon.trace**

### Default Log Files

The log files in *install\_dir\analysis\default\log* (*\$OV\_DATABASE/analysis/default/log*) are an audit trail of successful transactions since your last backup. These files are required for recovering the database at a point *other than* when the last backup was performed. These default log files are normally present with all relational databases.

If you are not performing backups on a regular basis, these log files could grow without bounds. If you are not concerned about recovering your database, you can remove these files at any time. New log files are constantly generated by your relational database software.

## Weekly Tasks

You may wish to perform the following tasks on a weekly basis.

### Back Up NNM

It is a good idea to do a full backup weekly (or as often as required at your site), so that you can restore to a relatively current state in the event of a system failure or a corrupted database. This will also ensure that you have all of the databases and configuration files together.

If you perform backups on a regular basis, you can restore to a relatively current state in the event of a system failure or a corrupted database.

There are a number of other situations when you should back up your NNM files. At a minimum, always do a full backup before starting any of the following tasks:

- Upgrading to a new version of NNM.
- Extensively editing or customizing your map.
- After performing initial or expanded discovery, once you think your map accurately reflects your network.

Refer to Chapter 6, “Preserve Your Sanity: Backup and Polling Configuration.”

### Check Polling Performance

You should periodically check to see how well netmon is keeping up with the changes in your network. Use `Performance:Network Polling Statistics` to gauge how effectively netmon is collecting network data. If netmon is not keeping up with the network changes, you can adjust polling intervals and/or queue sizes (see “Fine-Tuning the Polling Services” on page 178 for more information). You can also consider reducing the size of the managed environment (see “Automatically Limit Your Management Domain” on page 107 for more information).

## Check Web Launcher Log Files

The Web Launcher log files record errors and messages from the Launcher. These files will grow without bounds, and should be truncated as needed. The Launcher log files are:

*Windows:*

```
install_dir\www\logs\launcher\error_log  
install_dir\www\logs\login_log  
install_dir\www\logs\access_log
```

*UNIX:*

```
/var/opt/OV/www/logs/launcher/error_log  
/var/opt/OV/www/logs/launcher/login_log  
/var/opt/OV/www/logs/launcher/access_log
```

## Monthly Maintenance Tasks

You should perform the following tasks on a monthly basis.

### Check Patch Releases

Check the HP OpenView Web site for the latest patch releases or support updates. These can usually be downloaded at no charge. The HP OpenView Web site is:

<http://openview.hp.com>

### Check License Requirements

When running NNM 250, it is a good idea to check to see if you are going to need more licenses. If your network is growing quickly and you have added many nodes, you could run out of available licenses. When the system gets within 25 nodes of the licensed nodes limit, an application alert will be generated. However, you can also check the number of nodes being managed and the number of nodes licensed before this alert is generated. To check the status of licenses:

1. In the NNM main window, right-click on the Internet object.
2. Select Object Properties from the pop-up menu.
3. Select IP Map and click Edit:Attributes. This will display the number of nodes that you are licensed for and the number of nodes that are currently being managed.

## **Yearly Maintenance Tasks**

You should perform the following tasks on a yearly basis.

### **Evaluate Latest Release**

When new releases of Network Node Manager are announced, you should evaluate the new features that are included. If your network management needs have changed, the newest release may have new tools and features that will help you manage your network more effectively.

## Other Maintenance Tasks

There are other maintenance tasks that you may want to perform occasionally to improve the performance of the system, or just to make routine tasks easier.

### Enabling/Disabling Automatic Startup

The NNM installation service creates the appropriate files such that the host operating system automatically starts NNM when the management station boots. Occasionally you may want to disable automatic NNM startup, such as during system maintenance when the system is going to be rebooted several times.

#### On a Windows Operating System:

1. From the Windows Start menu, select Settings:Control Panel.
2. In the Control Panel, click on Services.
3. Select HP OpenView Process Monitor and choose Manual Startup.

#### On a HP-UX Operating System:

Edit the startup file `/etc/rc.config.d/ov500` and set the variable `START_OV500` to either 1, to start NNM at boot up or to 0, to disable NNM boot at startup. You do not need to reboot the system after making this change.

#### On a Solaris Operating System:

To disable NNM from starting at boot time, rename the filename `S98netmgt` within the sequencing link `/etc/rc3.d/S98netmgt` to a name that begins with a letter other than “S” or “K”. To re-enable NNM starting at boot time, change the filename back to `S98netmgt` and reboot.

#### On a Linux Operating System:

To disable NNM from starting at boot time, rename the files `K01netmgt` and `S98netmgt` within the directory `/etc/rcn.d/` (where *n* is the run-level in which the system is rebooting) to a name that begins with a letter other than “S” or “K”. To re-enable NNM starting at boot time, change the filename back to the original filenames and reboot.

## Customizing the Startup Configuration

The service `ovspmd` starts each service according to instructions it finds in the startup file `install_dir\conf\ovsuf` (`$OV_CONF/ovsuf`). Depending on which services are configured to start, system response time can be impaired, performance may be slower, and disk space can be rapidly consumed. If you encounter these problems, consider modifying the startup configuration.

Do not edit the `ovsuf` file directly. Instead, configure startup options in the local registration file (LRF) for the individual services. LRF files are then submitted to `ovsuf` through the `ovaddobj` command. Conversely, service startup instructions are disabled from `ovsuf` using the `ovdelobj` command.

Table 15-1 describes each LRF and any options applied as defaults in the LRF. The table does not contain a complete list of options. To learn about all the available options, see the reference page in NNM's online help (or the UNIX manpage) with the particular service's name.

The steps for customizing the startup configuration follow Table 15-1.

**Table 15-1** Local Registration Files (LRFs)

LRF	Description
netmon.lrf	<p>The LRF for <code>netmon</code>, the background service that polls agents to initially discover your network.</p> <p>Default option: <code>-P</code> which means that it is started through <code>ovstart</code>.</p> <p>You may want to use the following options. Be aware, however, that using these options can affect performance.</p> <p><code>-b burstcount</code></p> <p>Add if you have an unreliable network. This allows you to do multiple pings on a node.</p> <p><code>-s seedfile</code></p> <p>Add when you want the initial map to include specific networks in your administrative domain.</p> <p><code>-J</code></p> <p>Add if you have few SNMP agents on your network to accelerate initial discovery.</p>



**Table 15-1 Local Registration Files (LRFs) (Continued)**

LRF	Description
ovactiond.lrf	<p>The LRF for ovactiond, the background service that executes configured commands when an event is received.</p> <p>Default option: none.</p>
ovcapsd.lrf	<p>The LRF for ovcapsd, the background service that discovers Remote DMI nodes, Web-manageable nodes, and Web servers.</p> <p>Default option: none.</p>
ovrepld.lrf	<p>The LRF for ovrepld, the background service that replicates objects.</p>
ovtopmd.lrf	<p>The LRF for ovtopmd, the background service that controls the topology database.</p> <p>Default option: -O which means that it is started through ovstart.</p> <p>Use the -f <i>filter</i> option to specify a topology filter and make use of NNM's scalability and distribution feature.</p>
ovtrapd.lrf	<p>The LRF for ovtrapd, the background service that executes configured commands when an event is received.</p> <p>Default option: none.</p>
ovuispmd.lrf	<p>The LRF for ovuispmd, the background service that manages user interface services and distributes relevant ovspmd requests to OVW.</p> <p>Default option: none</p> <p>Use the -t option to change the time period that ovuispmd will wait for commands to complete.</p>
ovwdb.lrf	<p>The LRF for ovwdb, the background service that controls the object database.</p> <p>Default option: -O which means that is started by ovstart.</p> <p>You may want to use the -n <i>number of objects</i> option to reduce the amount of memory used by ovwdb when managing large numbers of nodes.</p>

**Table 15-1 Local Registration Files (LRFs) (Continued)**

LRF	Description
pmd.lrf	The LRF for pmd, the background service that multiplexes and logs events. Default option: none.
snmpCollect.lrf	The LRF for snmpCollect, the background service that collects MIB data. Default option: none.

### Steps for Customizing the Startup Configuration

1. Stop NNM.
  - a. Exit any currently-running `ovw` sessions.
  - b. Run `ovstop -c`
2. Create a back-up copy of the service's LRF file.
3. Open the selected service's LRF file in a text editor:
4. Edit the lines that control the behavior you want to change. Save and close the file.
5. Submit the new LRF file with the `ovaddobj` command:  
`ovaddobj $OV_LRF/netmon.lrf`
6. Restart NNM. Run `ovstart -c`

---

# **A** **NNM Services and Files**

This appendix describes key services (processes) and files in the operation of Network Node Manager, but does not include service and file updates for the operation of the Extended Topology functionality.

Topics covered include:

- Differences between background services and foreground services (page 525).
- How the services interact at NNM startup and during operation (page 531).
- The web application services and how they interact (page 529).

---

## Services and Files

There are two types of services:

- Background services that run continuously independent of whether `ovw` is running.
- Foreground services that run while `ovw` is running.

The interaction among these services are shown in the figures in this appendix. A brief description of each service and file follows. For more information on these services and files, see the reference page in NNM's online help (or UNIX manpage) having the same name as the service you wish to learn about.

### Background Services

The following services and files are involved with the startup of NNM.

<code>ovstart</code>	Starts the various services that make up the HP OpenView Network Node Manager product.
<code>ovspmd</code>	Launches and manages all background services. <code>ovspmd</code> interacts with the user commands <code>ovstart</code> , <code>ovstop</code> , <code>ovstatus</code> , <code>ovpause</code> , and <code>ovresume</code> , and performs the appropriate actions on the background services.
<code>ovsuf</code>	Contains the configuration information for <code>ovspmd</code> . Each entry in <code>ovsuf</code> is created by <code>ovaddobj</code> from information in the LRF (Local Registration File).

The following services run in the background during regular operation. All of the following files reside in `install_dir\bin\($OV_BIN)`.

<code>httpd</code>	Handles <code>http</code> requests. This is a web server that is provided on the UNIX operating system only.
<code>netmon</code>	Polls SNMP agents to discover your network, and then detects topology, configuration, and status changes in the network.
<code>ovactiond</code>	Receives events from <code>pmd</code> and executes commands.

<code>ovalarmsrv</code>	Provides event information to Java-based Alarm Browsers.
<code>ovcapsd</code>	Listens for new nodes and checks them for remote DMI capabilities, web-manageability, and web server capabilities.
<code>ovas</code>	Maintains topology and node status information for NNM Dynamic Views.
<code>ovrepld</code>	Responsible for object replication. <code>ovrepld</code> is started only when monitoring remote collection stations in a DIDM environment.
<code>ovrequestd</code>	Executes the reports and data warehouse exports according to a predefined schedule. Once a report is configured, <code>ovrequestd</code> starts executing the exports and the reports.
<code>ovsessionmgr.exe</code>	Manages users' web sessions. <code>ovsessionmgr.exe</code> is started by <code>ovstart</code> .
<code>ovtopmd</code>	Maintains the network topology database. The topology database is a set of files that stores <code>netmon</code> polling status and information about network objects, including their relationships and status. <code>ovtopmd</code> reads the topology database at start-up.
<code>ovtrapd</code>	Receives SNMP traps and forwards them to <code>pmd</code> . <code>ovtrapd</code> also responds to SNMPv2 requests.
<code>ovuispmd</code>	Manages the NNM user interface services and distributes relevant <code>ovspmd</code> requests to each instance of <code>ovw</code> that is running. <code>ovuispmd</code> must be running for <code>ovw</code> to be started, and should be running whenever <code>ovspmd</code> is running.
<code>ovwdb</code>	Controls the object database. The object database stores semantic information about objects.
<code>pmd</code>	Receives and forwards events, and logs events to the event database. <code>pmd</code> also forwards events from the network to other applications that have connected to <code>pmd</code> using the SNMP API.

`snmpCollect` Collects MIB data and performs threshold monitoring. `snmpCollect` stores the data it collects in the `install_dir\databases\snmpcollect` (`$OV_DB/snmpcollect`) directory and sends threshold events to `pmd`.

## Foreground Services

This section describes the NNM foreground services that may be running after you execute `ovw`. All of these files reside in `install_dir\bin\` (`$OV_BIN`):

`ipmap` Runs under `ovw` to automatically draw IPX and IP topology maps representing your network.

`mibform`, `mibtable`, `rnetstat`, `findroute`, `rbdf`, `rnetstat`, `rping`  
Commands executed by `xnmappmon` that enable you to monitor and diagnose problems in your TCP/IP network.

`ovw` The service that provides map drawing, map editing, and menu management. When you start the OVV services by executing the `ovw` command, `ovw` automatically invokes `ipmap`, `xnmevents`, and the other applications, which register to be started by `ovw`.

`xnmappmon` The Application Encapsulator that displays textual results of monitoring operations for managed SNMP objects selected from the map.

`xnmbrowser` The Tools:SNMP MIB Browser menu item that enables you to get and set MIB values for Internet-standard and enterprise-specific MIB objects.

`xnmbuilder` The Options:MIB Application Builder:SNMP menu item that enables you to build custom screens to manage multivendor MIB objects. The information you define using the MIB Application Builder is stored in registration files and help files using `mibform`, `mibtable`, and `xnmgraph`.

`xnmcollect` The Options>Data Collection & Thresholds:SNMP menu item that enables you to configure `snmpCollect` to collect MIB data from network objects at regular, configurable intervals. The configuration information

is stored in the *install\_dir*\conf\snmpCol.conf (\$OV\_CONF/snmpCol.conf) and snmpRep.conf configuration files. The collected data are stored in files in the *install\_dir*\databases\snmpCollect (\$OV\_DB/ snmpCollect) directory.

xnmevents	The alarm browser that is automatically invoked by <code>ovw</code> to display events that are being received by <code>pmd</code> . <code>xnmevents</code> reads the <i>install_dir</i> \log\xnmevents (\$OV_LOG/xnmevents). <i>username.map</i> file only at start-up for events that occurred since the last time <code>xnmevents</code> was run. <code>xnmevents</code> reads the event database only at start-up to obtain those pending events. <code>xnmevents</code> also reads <code>trapd.conf</code> to obtain information about how to customize event messages.
xnmgraph	The tool that enables you to graph the results of monitoring operations for managed SNMP objects selected from the map. The results may be real-time or collected historical data.
xnmloadmib	The Options:Load/Unload MIBs:SNMP menu item used to load new Internet-standard or enterprise-specific MIBs into the loaded MIB database.
xnmpolling	The Options:Network Polling Configuration->IP/IPX menu item that updates the polling configuration.
xnmsnmpconf	The Options:SNMP Configuration menu item that enables you to add, delete, and modify SNMP configuration parameters and the <code>netmon</code> status polling interval. The SNMP configuration parameters include community name, set community name, timeout interval, number of retries, and proxy information.
xnmtrap	The event configurator invoked by the Options:Event Configuration menu item that enables you to define enterprise-specific events (traps). You can customize the alarm message displayed through <code>xnmevents</code> when a particular event arrives. You can also specify a command or a batch file that should be executed when



a particular event arrives. Event configuration changes are stored in the *install\_dir*\conf\C\trapd.conf (\$OV\_CONF/\$LANG/trapd.conf) configuration file.

## Web CGI Programs

The following programs are CGIs, which access HP OpenView's web applications. See the *OVWebURLIntro* reference page in NNM's online help (or the UNIX manpage) for more information about (\*) only. The others are used internally by NNM and listed only for your information.

<code>jovw.exe</code>	*	Starts the Network Presenter.
<code>jovwreg.exe</code>		Called by the Network Presenter to parse menu information from application registration files registered with the Network Presenter. Also parses symbol and enroll information from registration files registered with <code>ovw</code> .
<code>nmmRptConfig.exe</code>	*	Launches the Report Configuration interface.
<code>nmmRptPresenter.exe</code>	*	Launches the Report Presenter.
<code>ovalarm.exe</code>	*	Launches the Alarm Browser user interface.
<code>ovlaunchreg.exe</code>		Reads the Launcher registration files (WLRFs), passing input to the Launcher window for display.
<code>ovlaunch.exe</code>	*	Launches the HP OpenView Launcher and opens the user login screen, when <code>UserLogin</code> is set.
<code>ovlogin.exe</code>		Processes the user login screen. Should not be called as a standalone URL.
<code>ovsessioninfo.exe</code>		Collects session information from the ongoing session.
<code>ovwebdata.exe</code>		Tool for accessing NNM data stores via read-only http queries.
<code>OvHelp</code>		The NNM 5.0 help CGIs.
<code>OvWebHelp.exe</code>		The NNM 6.0 help CGIs.
<code>printsession.exe</code>		Browser-callable URL to present user session information.

`snmpviewer.exe` \* Launches the SNMP Data Presenter, or displays textual or graphical results of SNMP monitoring operations on your browser.

`webappmon.exe` Application encapsulator used by SNMP Data Presenter.

Figure A-4 on page 535 illustrates how the CGI programs interact with NNM's other services.

## Behavior of the Services and Files

Figure A-1 shows the interactions and relationships among background services and files at start-up time. The arrows indicate the direction that communication flows.

**Figure A-1** Services and Files at Start-up

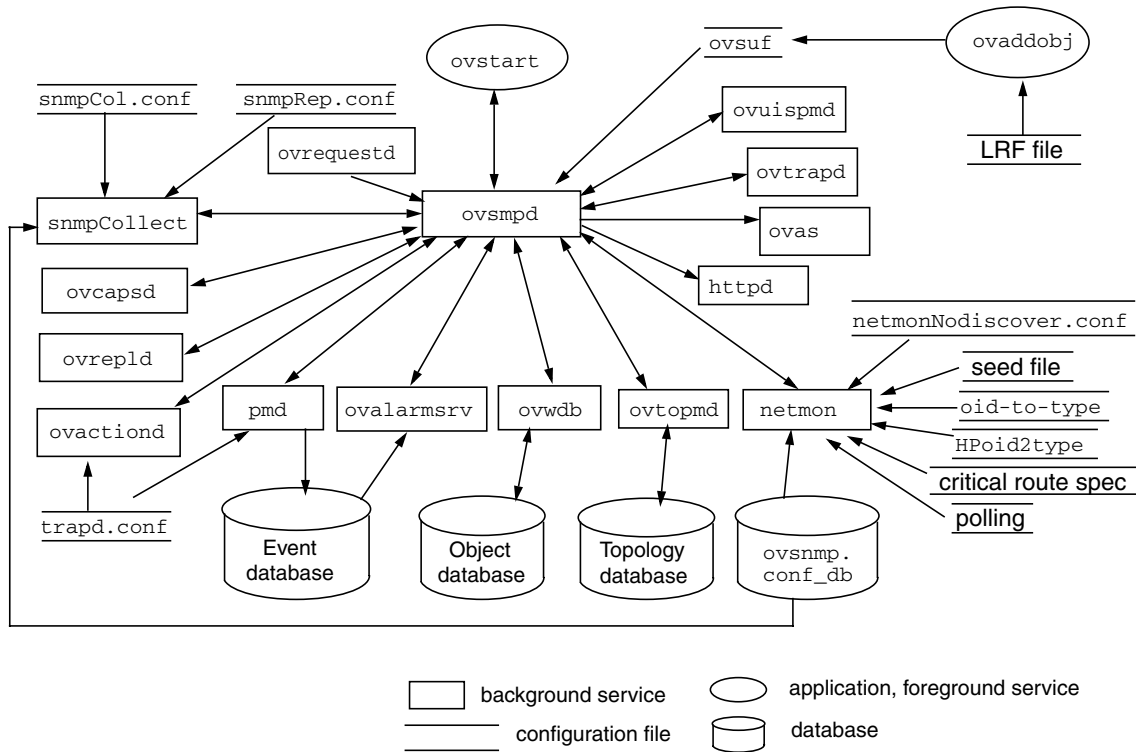


Figure A-2 and Figure A-3 show the interactions and relationships of background services and files during operation. The arrows indicate the direction that information flows.

**Figure A-2 Services and Files During Operation (Part 1)**

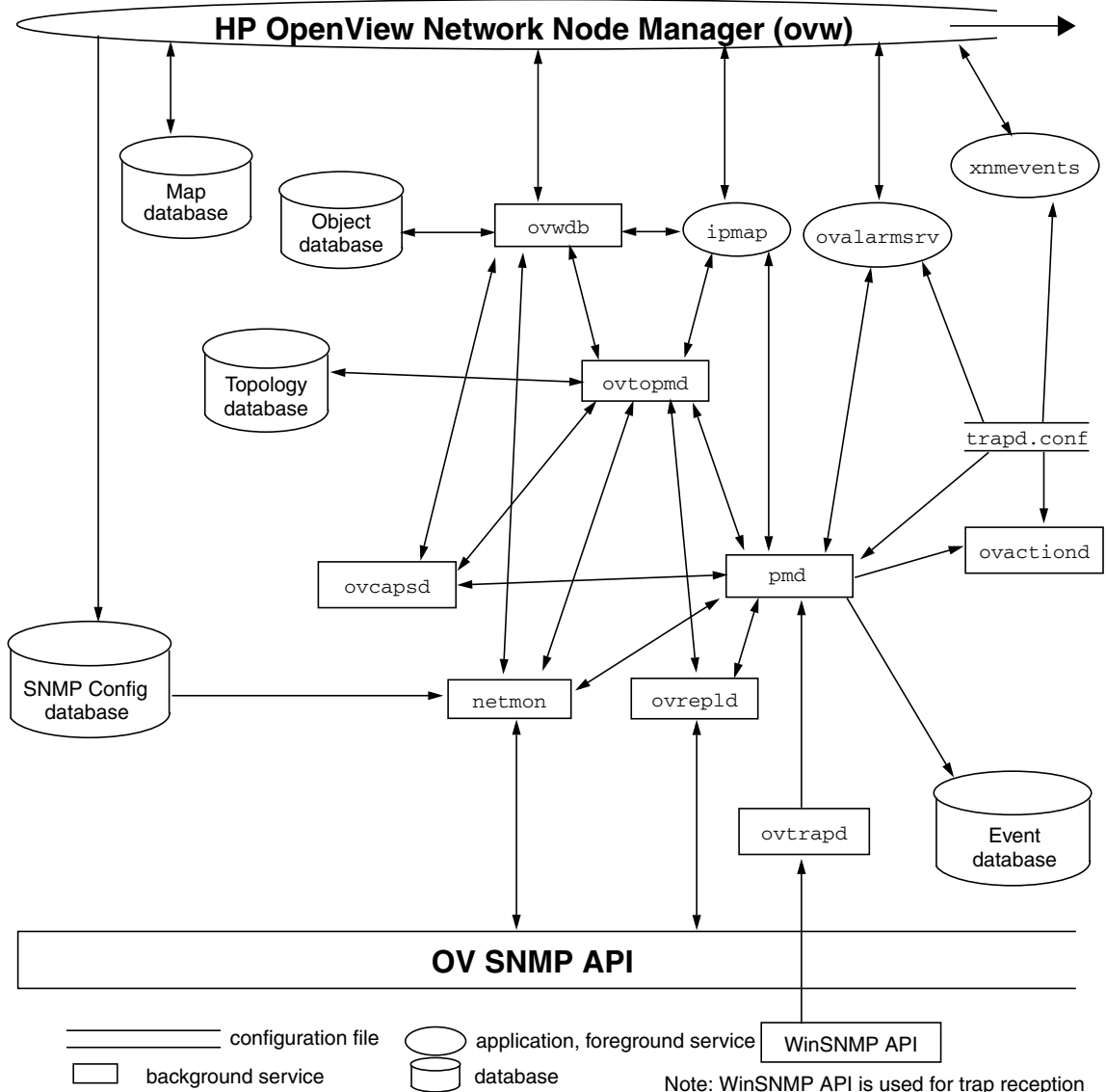


Figure A-3 Services and Files During Operation (Part 2)

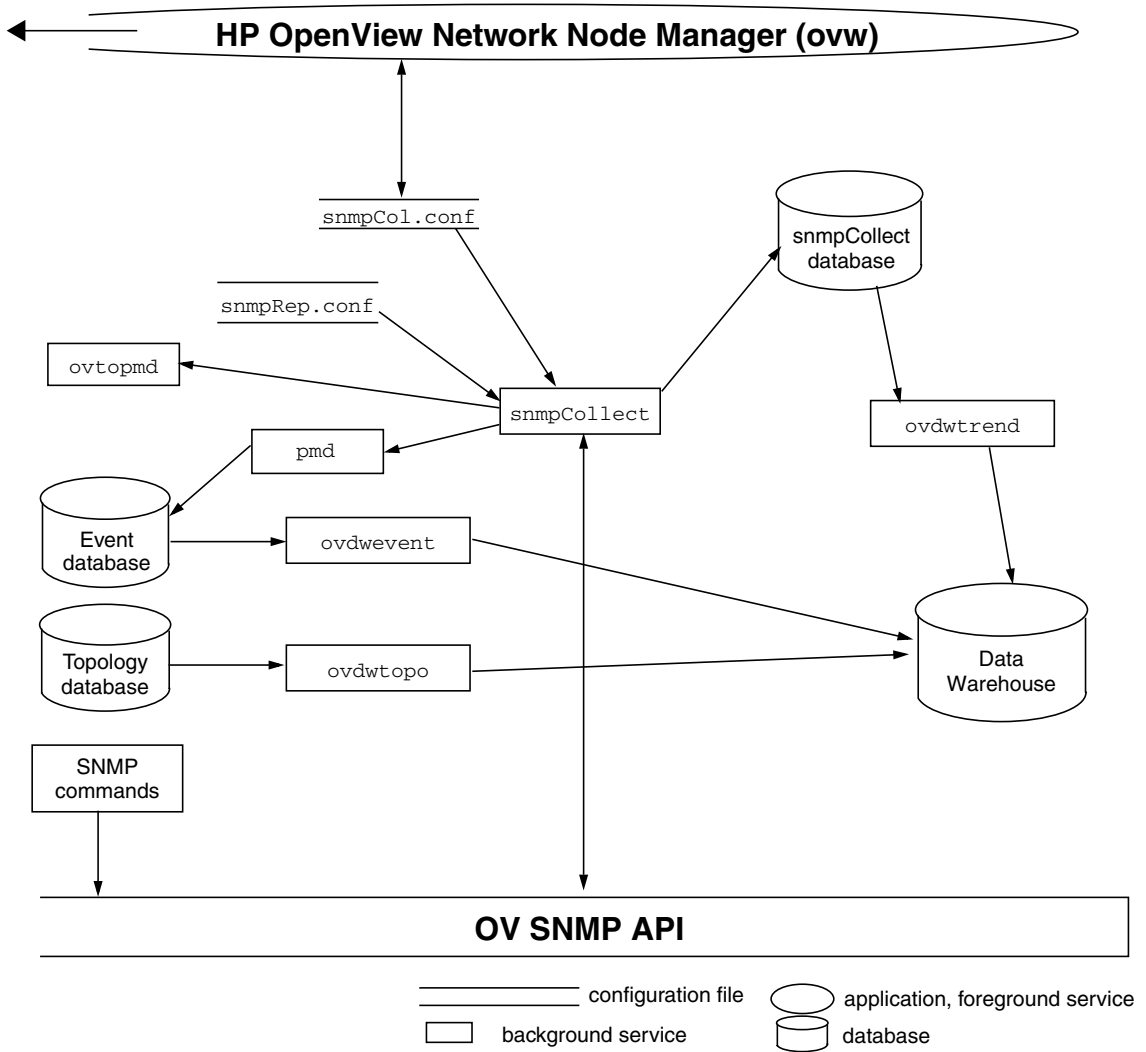
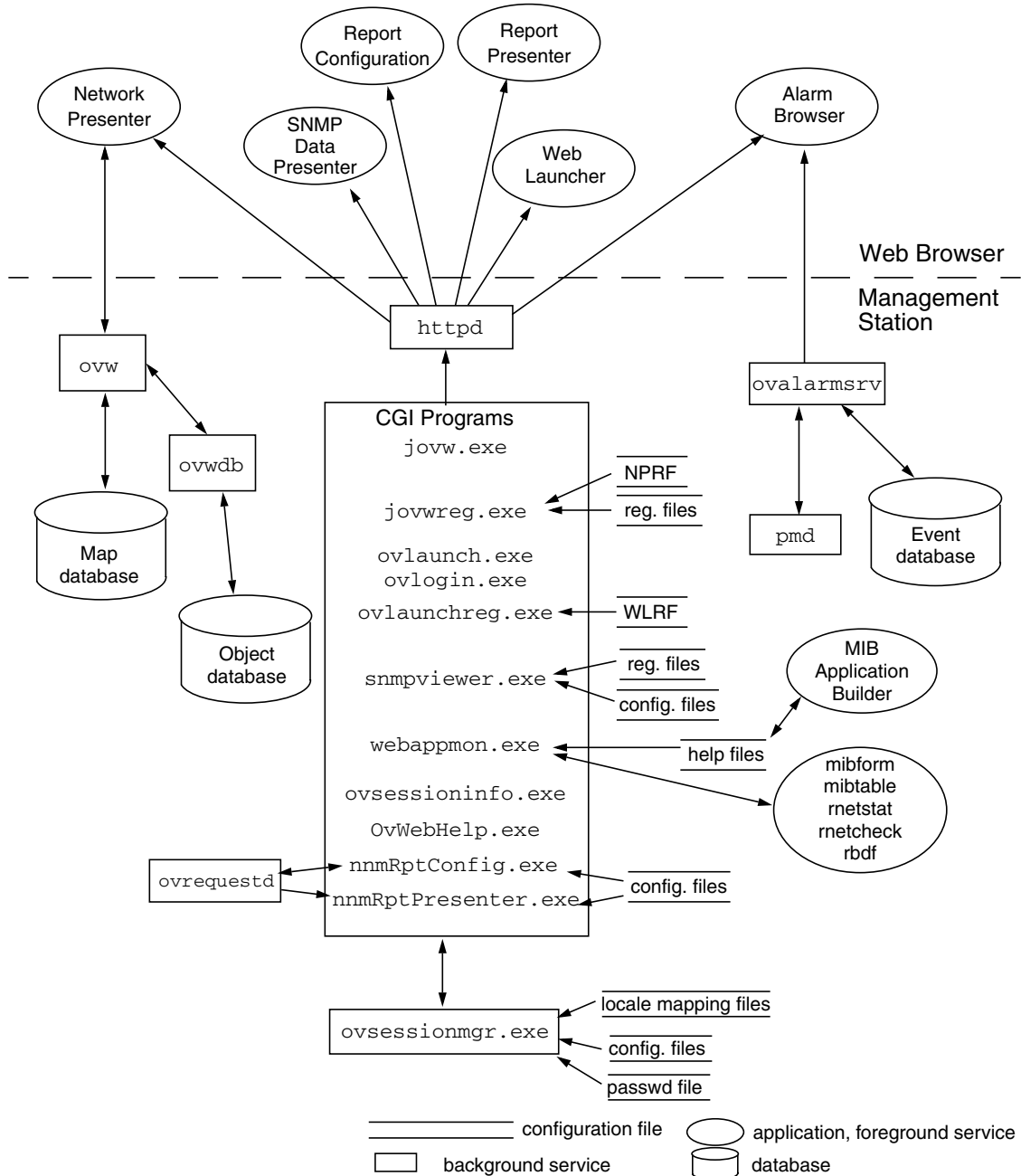


Figure A-4 shows the interactions and relationships among the services and files of the web components.

Figure A-4 Services and Files for the Web Components



NNM Services and Files

**Behavior of the Services and Files**



---

# **B Troubleshooting NNM Itself**

This appendix describes measures that you can use to identify and troubleshoot system and software problems that may occur during Network Node Manager operations. These measures include the following:

- General network management troubleshooting considerations (page 539).
- Characterizing various types of problems (page 542).
- Troubleshooting background services (processes), such as `netmon` and `ovtopmd` (page 544).
- Troubleshooting the Web applications (page 576).
- Troubleshooting various operations, such as MIBs and runtime components (page 586).
- Troubleshooting Windows operating system applications (page 600).
- Recommended practices for logging and tracing (page 602).
- Improving performance and traffic management (page 617).

These measures are not intended to be used for troubleshooting your network problems, but are only for system and software problems related to NNM operations. These measures are not intended to be used for troubleshooting problems with Extended Topology functionality.

Most of the steps you will take in troubleshooting NNM are the same for NNM on either a Windows or UNIX operating system; however, some files and services are different. In addition, there are considerations to NNM on Windows operating systems that do not apply to NNM on UNIX operating systems, and vice versa.

---

**NOTE**

For troubleshooting NNM's databases and the data warehouse, refer to the online manual *Reporting and Data Analysis with HP OpenView Network Node Manager*.

---

## General Troubleshooting Considerations

Consider the following when troubleshooting NNM:

- Network Node Manager is memory and page-file (swap-space) intensive. Problems may be due to resource exhaustion. Refer to “Performance” on page 619 for memory conservation tips.
- If you are using the Berkeley Internet Name Domain (BIND) Service on your network, each `gethostbyaddr` request for the hostname of a host with multiple IP addresses could return a different hostname.

## When You Need More Information

Other information that may assist your troubleshooting is available in the following locations:

- The release note files (see page 53).
- The NNM online help.

If a problem does not appear to be with the network management products themselves, see the following:

- Networking documentation for general network troubleshooting procedures.
- System documentation for system troubleshooting procedures.
- On UNIX operating systems only, X Windows documentation for X Windows troubleshooting procedures.

## Preventive Practices

Following these recommended practices can help you prevent problems:

- Ensure that the management station meets the hardware, software and configuration prerequisites and recommendations.
- Maintain the system by regularly editing and deleting files. See Chapter 15 for maintenance tasks.
- Ensure that `install_dir\bin\` (`$OV_BIN`) is in your path.

- On a UNIX operating system, source `/opt/OV/bin/ov.envvars.sh` into your environment.
- On a Windows operating system, source `install_dir\bin\ov.envvars.bat` into your environment.
- Do not modify Network Node Manager product files without retaining the original files. The original files provide a way of restoring a known good operational configuration. If you correct a problem by restoring the original files, you can isolate the problem to one of the changes you made to these 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 will be used and you will end up with duplicate definitions.

```
install_dir\registration\ ($OV_REGISTRATION)
install_dir\fields\ ($OV_FIELDS)
install_dir\symbols\ ($OV_SYMBOLS)
install_dir\bitmaps\ ($OV_BITMAPS)
install_dir\conf\ ($OV_CONF)
install_dir\help\ ($OV_HELP)
install_dir\tmp\ovbackup ($OV_TMP/ovbackup)
```

See the backup procedure in Chapter 6 for complete guidelines and procedures for backing up your files.

- Make sure that you are not using up too much of your management station’s CPU and network resources by collecting too much MIB data or setting too frequent polling intervals for the netmon service. Use Performance:Network Polling Statistics to gauge how effectively netmon is collecting network data.
- Use logging and tracing of background services to help isolate problems, but be sure to clean up log and trace files regularly. For more information see “Recommended Logging and Tracing Practices” on page 602” in this appendix.
- Use `ovstatus -c -v` to check the status of background services and OVW sessions.
- Use `ovstart -c` to track the progress of the services being started by `ovspmd`.
- Use the `-d` option to `ovstart`, `ovstop`, `ovpause`, and `ovresume` to get additional debugging information.

- Use the `-verify` option to `ovw`. This option checks the syntax for all the installed application, symbol type, and field registration files. It also verifies the semantics of the application registration files and reports conflicts among application registration information. When started with this option, NNM does not start a normal “windowed” session; rather, it displays status messages in a text window.
- Use the `-mapcount` option to `ovw`. This option checks the consistency between the map database maintained by NNM and the object database maintained by `ovwdb`.
- Be careful when changing the permissions on files used by the management software. Restricting access could cause management services to be unable to access the file.
- On UNIX operating systems only, HP recommends that you not modify the files in `$APP_DEFS`. Instead, you can override the default X resources by modifying them in your `$HOME/.Xdefaults` file. If you are running HP VUE, modify your VUE resources. See `$APP_DEFS/XNm` for more details.

## Characterizing the Problem

Symptoms are visible conditions that indicate a problem. Whenever you encounter a symptom, collect the basic information described in the following subsections. The problem can usually be characterized in one or more of the following ways: scope of the problem, what changed just before the problem occurred, duration of the problem, and what action was performed when the problem occurred.

### Distinguish Local versus Remote Problems

If you suspect a problem on a remote node or agent system, try to duplicate it on a local node or management station, and vice-versa. If you cannot duplicate the problem on the local node, the problem is probably associated with the remote node. If you can duplicate the problem on the local node, the problem may be in the management station or somewhere in the link, or may be a problem common to the agents.

### Remote Nodes' Level of Manageability

What type of remote node are you trying to communicate with when the problem occurs?

The amount of information that NNM can collect depends on whether a remote node:

- Supports SNMP or not (or is proxied).
- Supports all the mandatory objects in MIB-II.
- Supports a partial, full, or extended (non-standard) MIB.
- Is an HP node or not.
- Is a TCP/IP node or not.
- Is an IPX node or not.
- Is an IPX router/server or not.

The management station gets the most information from an HP SNMP node, less information from other SNMP nodes, least information from non-SNMP and TCP/IP nodes, and no information from non-SNMP and non-TCP/IP nodes.

## **Context: What Changed?**

Determine what may have changed recently on your network or product configuration: hardware, software, files, security, utilization, etc. Keep a record of the software patches you install, so you can undo the changes in the order they occurred.

Also determine what is unique about a particular node that is having problems.

## **Duration: How Long or How Often?**

Is the problem consistent (fails every time) or intermittent (fails sometimes)?

## **Context: What Action Was Performed?**

When the problem occurred, what was happening? For instance:

- What operation was selected?
- What command was executed?
- What mouse operation was performed?
- What data was requested or sent?
- Was an alarm generated?
- Was an error or warning logged through the Windows Application Event Log (or `nettl`)?
- Was an error or warning logged through `ovconsole`, or the window from which `ovw` was started?
- Does `ovstatus -c` show all background services running?
- Does `ovstatus -v ovuispmd` list all of the OVW services that are expected to be running?

## Troubleshooting Background Services

This section explains the background services of NNM and suggests actions to take if you have problems with one of the following background services. The background services are:

- `hpptd`
- `netmon`
- `ovactiond`
- `ovalarmsrv`
- `ovcapsd`
- `ovrepld`
- `ovrequestd`
- `ovtopmd`
- `ovtrapd`
- `ovuispmd`
- `ovwdb`
- `pmd`
- `snmpCollect`

In addition to explaining the background services, this section explains how to troubleshoot `ovspmd`, the service that monitors and controls the background services.

The background services and `ovspmd` should always be running whenever NNM is running. Normally, the background services are started through `ovstart`. See the *ovstart* reference page in NNM's online help (or the UNIX manpage) for all the options to `ovstart`. It may be particularly useful to use the `-c` or the `-d` options when troubleshooting.

To verify that the background services are running, enter the following:

```
ovstatus -c
```

See the *ovstatus* reference page in NNM's online help (or the UNIX manpage) for all the options to `ovstatus`.

The result should look similar to the following:



Name	PID	State	Last Message(s)
OVsPMD	3676	RUNNING	-
ovsessionmgr	3677	RUNNING	Initialization complete.
ovwdb	3678	RUNNING	Initialization complete.
ovuispmd	3701	RUNNING	Initialized. 1 ovw clients registered.
ovtrapd	3698	RUNNING	Initialization complete.
ovactiond	3699	RUNNING	Initialization complete.
ovalarmsrv	3700	RUNNING	Initialization complete.
pmd	3679	RUNNING	Initialization complete.
ovdbcheck	3680	RUNNING	Connected to embedded database.
ovrequestd	3681	RUNNING	Initialization complete.
httpd	-	unknown	(Does not communicate with ovspmd.)
ovtopmd	3697	RUNNING	Connected to native database "openview".
netmon	3702	RUNNING	Initialization complete.
snmpCollect	3703	RUNNING	No values configured for collection.

If `ovstatus` indicates that an object manager was not found, the problem may be caused by one of the following:

- The `ovspmd` configuration has been incorrectly modified or is corrupted. If the `ovspmd` configuration is corrupted, see “The `ovspmd` Background Service” on page 554.
- The executable file has been removed or moved from its expected location. `ovsuf` either has a pathname to each executable or expects them to be in `install_dir\bin\ ($OV_BIN)`. Make sure they exist where they are supposed to be.

If `ovstatus` indicates that a service is in a paused state, it may mean that a backup is currently being performed, or that the service did not restart as expected after a backup. In that case, the state will be one of the following: `PAUSED`, `RESUME_ERROR`, or `DEPENDENCY_ERROR`. In this case, check the backup log in `install_dir\tmp ($OV_TMP)` to determine the cause of the error. You can then stop all services using `ovstop`, and restart all the services by typing:

**ovstart -c**

Some of the background services depend on other services to operate properly. Table B-1 shows the dependencies.

**Table B-1 Background Service Dependencies**

Background Service	Dependencies
httpd	no dependencies

**Table B-1 Background Service Dependencies**

<b>Background Service</b>	<b>Dependencies</b>
netmon	ovwdb, pmd, ovtopmd
ovactiond	pmd
ovalarmsrv	pmd
ovcapsd	ovwdb, pmd, ovtopmd
ovdbcheck	none
ovrepld	pmd, ovtopmd
ovrequestd	none
ovtopmd	ovwdb, pmd, snmpd
ovtrapd	pmd
ovuispmd	ovwdb, ovtopmd
ovwdb	no dependencies
pmd	snmpd
snmpCollect	ovwdb, pmd, ovtopmd

For problems with individual background services, see the appropriate subsection in this appendix on how to troubleshoot a specific background service.

## **Associating Background Service Names and Port Numbers**

The installation process adds entries to the following file, associating official OpenView service names and aliases with specific port numbers and protocols:

- Windows: %SystemRoot%\system32\drivers\etc\Services
- UNIX: /etc/services

If you are using NIS (Network Information name Service) instead of the services file, you must update your NIS server with the appropriate OpenView service name to port association.

Table B-2 shows a list of many OpenView processes and their specific port numbers and protocols.

**Table B-2 OpenView Processes and Port Associations**

<b>Service Name</b>	<b>Port/Protocol</b>	<b>Service Description</b>
netwkpathengine	3209/tcp	HP OpenView Network Path Engine Server
ovalarmsrv	2953/tcp	OpenView Alarm Server daemon listener port
ovalarmsrv_cmd	2954/tcp	OpenView Alarm Server daemon command port
ovas	7510/tcp	HP OpenView Application Server
ovbus	7501/tcp	HP OpenView Bus Daemon
ovembeddb	2690/tcp	OpenView Embedded DW Database
ovhttp	3443/tcp	OpenView Web Server
ovsessionmgr	2389/tcp	OpenView Web Session Manager
ovspmd_mgmt	8886/tcp	OpenView Process Manager
ovspmd_req	8887/tcp	OpenView Process Manager
ovtopmd	2532/tcp	OpenView IP Topology daemon
ovuispmd	7777/tcp	OV UI Services Daemon
ovwdb	2447/tcp	OpenView Object Database daemon
pmdmgr	696/udp	OpenView Postmaster Manager

**Table B-2 OpenView Processes and Port Associations (Continued)**

Service Name	Port/Protocol	Service Description
snmp	161/udp	Simple Network Management Protocol Agent
snmp-trap	162/udp	Simple Network Management Protocol Traps

### The httpd Background Service

The httpd service handles http (web) requests. It is provided only on UNIX operating system installations of NNM. Here are some problems that can be caused by httpd and possible solutions to those problems.

httpd should be running wherever NNM is running. Normally it is started by ovstart. To verify that it is running, type:

```
ovstatus httpd
```

You can also see if httpd is running by typing `ps -elf | grep httpd` on UNIX operating systems. You should see multiple copies of httpd running, since the parent httpd service spawns multiple children to improve performance in handling httpd requests.

If httpd is running, you can validate that it is running correctly by entering the following into your web browser:

```
http://hostname:3443/
```

You should see a short web page with the title “It Worked!”

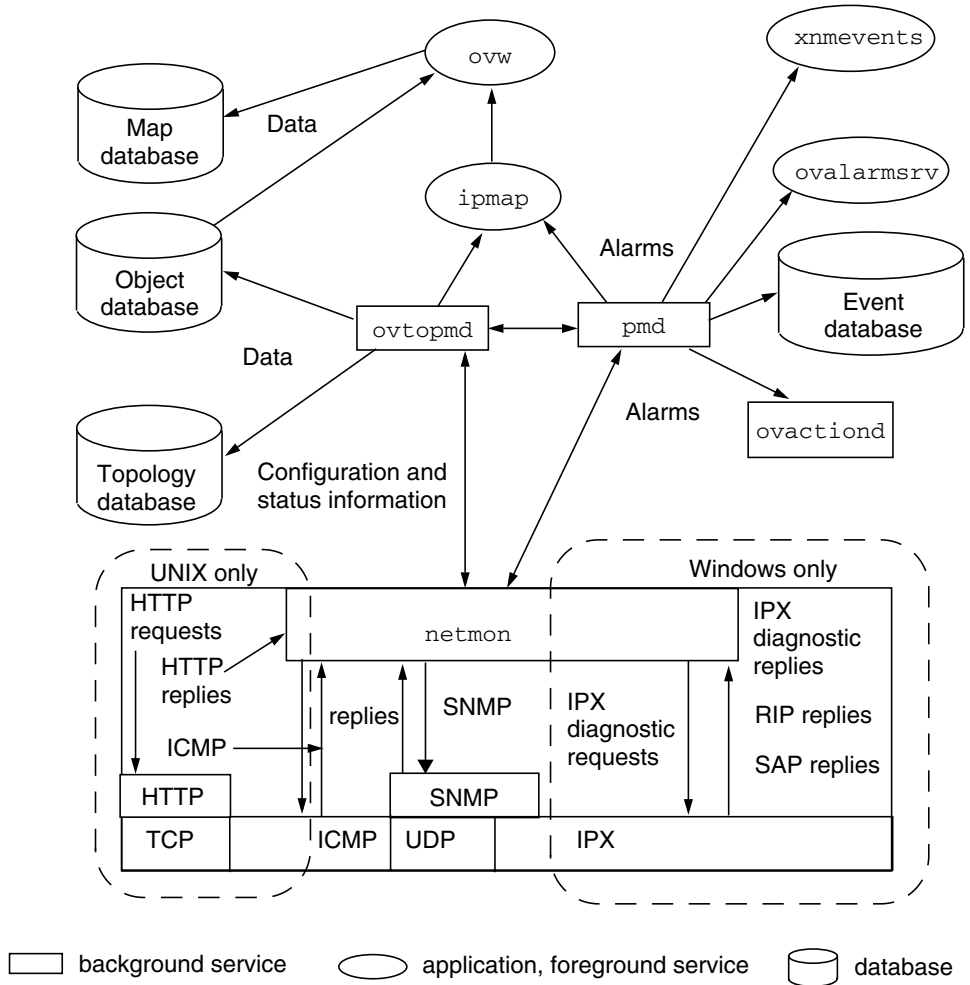
httpd maintains two log files. One is used to log errors (`httpd_error_log`) and the other records all http requests (`httpd_access_log`). You can check these log files to see if errors occurred during startup. The files are located in `$OV_PRIV_LOG`.

### The netmon Background Service

The netmon service uses IPX diagnostic requests and various ICMP requests (ping) to poll nodes for status, and SNMP requests to poll nodes for MIB values. When netmon detects a change, the service updates the topology database (through `ovtopmd`) with the change and sends the appropriate event to `pmd`. When configured to do so, `pmd` then logs the

event to the event database and forwards it to `ipmap`, `ovactiond`, and `xnmevents` (and potentially other subscribing applications). `ipmap` gets the information from `ovtopmd` and updates the map and the map database. `xnmevents` updates the Alarm Browser accordingly. Figure B-1 illustrates these `netmon` interactions. The area within the dotted oval indicates interaction on a Windows or UNIX operating system only. On UNIX operating systems, `netmon` discovers `http` requests.

**Figure B-1 netmon Interactions**



You can control netmon's polling through the map's Options:SNMP Configuration and the Options:Network Polling Configuration:IP/IPX menu item. The SNMP Configuration menu item modifies the *install\_dir\conf\ovsnmp.conf\_db* (\$OV\_CONF/ovsnmp.conf\_db) database. The Network Polling menu item stores netmon's polling intervals in the

*install\_dir*\conf\polling (\$OV\_CONF/polling) file. Do not modify this file directly; use the command line options to the program `xnmpolling`.

The `xnmevents` service checks every five minutes to see if `netmon` and `snmpCollect` are running. If either service is not running, `xnmevents` will display a dialog box and report the inactive service. (For information on changing the five-minute default polling interval or on disabling this polling feature, see the System Registry on Windows operating systems, or the file `/usr/lib/X11/app-defaults/XNmevents` on UNIX operating systems.)

Status polling is scheduled on a per node basis, but is not deterministic. Therefore, polling of a node may not occur when scheduled if `netmon` gets behind due to its polling load for a particular time interval. Also, polling of manually added nodes and nodes changed from unmanaged to managed may not occur immediately after they are added or managed.

While you are unmanaging a network or segment during active discovery, a few managed nodes may appear on that network. To avoid this problem, do one of the following:

- Wait until the program finishes active discovery of the map.
- Turn polling off while unmanaging the network.

Only one instance of `netmon` may run on the management station; the *install\_dir*\conf\ (\$OV\_CONF) `netmon.lock` file enforces this restriction.

The following tools can help you troubleshoot `netmon` problems.

- Use the `Fault:Network Connectivity:Poll Node or Status Poll` menu item to test or cause immediate `netmon` polling activity to a node.
- Use `ovdumpevents` to check for `netmon` service errors.
- Use the `Performance:Network Polling Statistics` menu item to determine if `netmon` is able to perform its polling at the configured frequency. See the online help for an explanation of the output of this menu item.
- Use `netmon` tracing to help isolate problems. For information see “Logging and Tracing” on page 602, or see the *ovstatus* reference page in NNM’s online help (or the UNIX manpage).

### **Auto-configuration of SNMP Traps**

When `netmon` detects a node running an HP OpenView SNMP agent, it attempts to add the IP address of the NNM system to the trap destination table using an SNMP `set` request. In order for this to work, the NNM system must be configured with the correct SNMP `set` community name using `Options:SNMP Configuration`.

### **The `ovactiond` Background Service**

The `ovactiond` service is started when you start NNM. It receives events from `pmd` that have automatic actions defined. For each of these actions, `ovactiond` formats the command string, then passes it on for interpretation and execution.

When you configure events using the `Options:Event Configuration` menu item in NNM, you are configuring `ovactiond`. If the result of your configuration of events is not what you expected, you can check the status of `ovactiond`.

Use the command `ovstatus ovactiond` to see if the service is running. `ovactiond` requires that `pmd` is running, so you should also check the status of `pmd`.

You can check the log file `ovactiond.log` for error messages. The file is located in:

*Windows:* `install_dir\logs`

*UNIX:* `$OV_LOG`

### **The `ovalarmsrv` Background Service**

The `ovalarmsrv` service is started when you start NNM, and is the service that forwards events to `xnmevents` on NNM on a management station, and to NNM's web-based Alarm Browser. `ovalarmsrv` relies on `pmd` to be running; therefore, if you have problems starting `xnmevents` or the Alarm Browser, first check to see if `pmd` is running using `ovstatus pmd`.

You can also check the status of `ovalarmsrv` by using the `ovstatus` command. If `ovstatus` shows that `ovalarmsrv` is not running, and it will not start using `ovstart ovalarmsrv`, check to see if the service is running but `ovstatus` is not reporting it.



On UNIX operating systems, type

```
ps -elf | grep ovalarmsrv.
```

On Windows operating systems, use the Task Manager. If you find an `ovalarmsrv` service running, kill that service and restart `ovalarmsrv` using `ovstart ovalarmsrv`.

If you still have problems starting `ovalarmsrv`, be sure that you do not have a port conflict. `ovalarmsrv` uses two ports for communications. You can check those values in the `/etc/services` file. Make sure that another application is not using one of the ports assigned to `ovalarmsrv`.

`ovalarmsrv` reads alarms from the event database, receives new alarms from `pmd`, and maintains the state of all user modifications (for example, acknowledge, delete, etc.) to presented alarms. If the event database or the state file have somehow become corrupted, or if the file permissions have changed such that those file can't be read, you will not receive any data in `xnmevents` or the Alarm Browser.

You can also check the `ovalarmsrv` log file, `ovalarmsrv.trace` for error and tracing messages. The file is located in:

*Windows:* `install_dir\logs`

*UNIX:* `$OV_LOG`

## The `ovcapsd` Background Service

The `ovcapsd` service checks newly-discovered nodes for Remote Desktop Management Interface (remote DMI or RDMI) and Hypertext Transfer Protocol (HTTP) capabilities. It periodically rechecks all Normal-status nodes. Capabilities are stored in the `ovwdb` database. `ovcapsd` gets IP and IPX address information for nodes from `ovtopmd`.

See the `ovcapsd` reference page in NNM's online help (or the UNIX manpage) for the most complete information available. Here are two important troubleshooting tips:

- `ovcapsd` checks nodes' capabilities after `netmon` has added them to the topology database. It might not have checked a node before the node appears in your `ovwdb` database or your map. If you search on a capability such as `isHTTPSupported`, `ovcapsd` might not yet have had a chance to check the node in which you're interested. You can run `ovcapsd` from the command line to check individual nodes synchronously. Also, `ovcapsd` can be made to check nodes faster by

increasing the number of threads it creates to check the nodes. See the *ovcapsd* reference page in NNM's online help (or the UNIX manpage) for more information.

- See the *install\_dir\log\ovcapsd.log* file for *ovcapsd* service errors and on-exit messages.

## The ovspmd Background Service

The *ovspmd* service monitors and controls the NNM background services. These background services are designed to run as children of *ovspmd*. The *ovspmd* service should be running whenever the background services are running. Normally, *ovspmd* is started by *ovstart*.

To verify the status of the services, type:

```
ovstatus -c
```

If *ovstatus* provides output indicating that all services are running, the problem is not *ovspmd*. If you do not receive the expected output, either *ovspmd* has a communication problem or the *ovspmd* service is not running. If the output indicates that *ovspmd* is not running, start it using *ovstart ovspmd*.

## Determining ovspmd Problems

You need to determine whether the problem is related to *ovspmd* or to a background service. If you determine you have an *ovspmd* problem, you need to determine what type of problem it is. Here are some indications of *ovspmd* problems:

- If the *ovstatus* output indicates that one of the background services is not running, type

```
ovstart -v
```

This should start any service that is not running. If any service fails to start, an error message will appear, indicating which service is not running. If this happens, use the Task Manager on a Windows operating system.

On UNIX operating systems only, type **ps -ef** to list currently running services.

Do *one* of the following:

- If the output indicates a certain service is running, but `ovspmd` says the service is not running (using `ovstatus` or `ovstart`), `ovspmd` does not know that service is running. This problem can occur when `ovspmd` is shut down unexpectedly.
- If the output indicates that a certain service is not running, that service has a problem. See the troubleshooting section for that service.

When you run `ovstart`, the error message you receive about the service should provide information that will help resolve the problem.

- If you receive an `ovspmd` error message in the following format:

```
ovspmd:conf\ovsuf: error message
($OV_conf/ovsuf: error message)
```

you may have a corrupt `ovsuf` configuration file.

On UNIX operating systems only, go to the next section “Recreating the `ovspmd` Sockets Directory (UNIX operating systems only)”.

- On UNIX operating systems only, if you receive *one* of the following error messages

```
ovstart: ovspmd is not running.It may have failed to start.
ovstart: unable to contact ovspmd: error message
```

you need to verify that the `ovspmd` sockets directory exists. Go to “Recreating the `ovspmd` Sockets Directory (UNIX operating systems only)” below.

### Recreating the `ovspmd` Sockets Directory (UNIX operating systems only)

This section is for UNIX operating systems only. If you are using NNM on a Windows operating system, see the section “The `netmon` Background Service” on page 548.

Use this procedure when `ovstart` is unable to start or connect to the `ovspmd` service.

1. Verify that the `$OV_SOCKETS` directory still exists. `ovspmd` creates the `OVSPMD_MGMT` and `OVSPMD_REQ` socket files in this directory, which enable `ovspmd` to communicate with the other background services. If this directory no longer exists, recreate it by typing

```
mkdir $OV_SOCKETS
```

2. Find out if any of the background services are running by typing

```
ps -ef
```

3. Kill all the NNM background services that are running using the `kill` command. *Do not* use `kill -9`. For a list of the services, see “Troubleshooting Background Services” on page 544 in this appendix.

4. To restart the services and verify that they are starting properly, type

```
ovstart -c
```

## The ovrepld Background Service

`ovrepld` is a service that manages the discovery and monitoring of objects done on behalf of the local NNM Advanced Edition management station by remote collection stations. This remote topology information is merged with information obtained locally by `netmon` to create a more comprehensive topology database. If objects known to be present on remote collection stations are not appearing on your map, or are not being updated as expected, it is possible that `ovrepld` has stopped functioning.

Use `ovstatus ovrepld` to determine if `ovrepld` is running. If it is not, you can start it using `ovstart ovrepld`. In addition, `ovrepld` relies on `pmd` and `ovtopmd` to be running, so check to be sure that those services are also running.

`ovrepld` writes errors to a log file, `ovrepld.log`. The file is located in:

*Windows:* `install_dir\logs`

*UNIX:* `$OV_LOG`

## The ovtopmd Background Service

The `ovtopmd` service is responsible for maintaining the topology database that is used by `netmon` and `ipmap`. It provides both enforcement of rules for correctness of the IP topology, and caching of the data to improve the performance of the system.

`ovtopmd` should be running whenever NNM is running. Normally it is started through `ovstart`. To verify that it is up and running, type

**ovstatus ovtopmd**

You should see something like this:

```
object manager name:  ovtopmd
state:                RUNNING
PID:                 4119
last message:        Connected to native database "openview"
exit status:         -
```

If `ovstatus` gives an error similar to this:

```
ovspmd:install_dir\conf\ovsuf:object manager ovtopmd not found
($OV_CONF/ovsuf:object manager ovtopmd not found)
```

The problem may be that the `ovspmd` configuration is corrupted. If the `ovspmd` configuration is corrupted, see “The `ovspmd` Background Service” on page 554 in this appendix.

If `ovstatus` reports the following message:

```
Running. Lost connection to master agent.
```

This could indicate that `ovtopmd` is still executing, but has lost its connection to the master SNMP agent (`snmpd`). Usually this is because the master agent (`snmpdm`) exited or has been shut down.

Restart the master agent on the Windows operating system using the Services applet in the Control Panel. Start the SNMP EMANATE Master Agent and SNMP EMANATE Adapter for NT/2000 services.

On UNIX operating systems only, restart `snmpd` by executing

```
/usr/sbin/snmpd
```

If the master agent is executing, try executing the following commands:

```
ovstop ovtopmd
```

```
ovstart ovtopmd
```

If `ovstatus` knows about `ovtopmd`, but indicates that it is not running, enter the following:

```
ovstart -c
```

Normally, this should start `ovtopmd` and all background services on which `ovtopmd` depends. In particular, `ovtopmd` depends on `ovwdb` and `pm` being started. If `ovwdb` does not start, `ovtopmd` will not be able to

execute successfully. In addition to `ovwdb` executing, `ovtopmd` relies on the field definitions being loaded into `ovwdb`, which normally happens through the configuration scripts at installation time.

If `ovtopmd` gives an error indicating that it was unable to map a field name into a field ID, do the following:

- Verify that `$LANG` is set correctly and that the `install_dir\fields\C ($OV_FIELDS/$LANG)` directory exists and contains at least the following files: `ovw_fields`, `ip_fields`, `snmp_fields`, `topm_fields`, and `misc_fields`. The `$LANG` environment variable provides support for localizing native languages. If not otherwise defined, `$LANG` is assumed to be `C`, which means that no native language support is provided. (UNIX systems only: For more information on native language localization, see the *lang* manpage.)
- Type the following commands:

```
ovstop -c
ovstart ovwdb
ovw -fields
ovstart -c
```

This should load any necessary field definitions into `ovwdb`, and then start `ovtopmd` and `netmon` (as well as any other background services).

By default, `ovtopmd` uses a TCP socket bound to port 2532. If `ovtopmd` indicates an error attempting to bind to this port number because another application is already using it, follow these steps:

1. Make sure that you do not have another `ovtopmd` service using port 2532.

The most common reason for errors binding to the TCP socket is the presence of another `ovtopmd` service. If an `ovtopmd` service was started outside of `ovstart`, the `ovstop`, `ovstart`, and `ovstatus` commands will not recognize this service. To prevent this problem, always start all NNM background services with the `ovstart` command and stop them with the `ovstop` command.

If another application is using port 2532, continue with the next step.

2. Change the port number to any unused port by changing the entry for `ovtopmd` in `%SystemRoot%\system32\drivers\etc\services (/etc/services)`:

```
ovtopmd 2532/tcp # OpenView IP Topology daemon
```

Choose a number other than 2532 that does not conflict with other services listed in `services (/etc/services)`.

3. Reconfigure NNM by typing the following commands:

```
ovstop
```

```
ovstart
```

4. If `ovtopmd` still fails to run successfully from `ovstart`, contact your HP support representative.

---

**NOTE**

On UNIX operating systems only, if you are using the Network Information Service (NIS), make sure you update your NIS Master Server.

---

Once `ovtopmd` has been started through `ovstart`, you can verify that it is responding to requests by executing `install_dir/bin/ovtopodump ($OV_BIN/ovtopodump)`. By default, `ovtopodump` will print a single line of output for each object in the topology. You can also get verbose listings and summary information. For more information see the *ovtopodump* reference page in NNM's online help (or the UNIX manpage). For example, if you type

```
ovtopodump -l
```

or, for remote collection stations

```
ovtopodump -c nodename
```

your result will look something like

```
TOPO OBJECT ID: 70  
TIME CREATED: Fri Apr 10 14:52:02 1995  
TIME MODIFIED: Fri Apr 10 14:52:02 1995  
GLOBAL FLAGS:  
NUMBER OF NETWORKS: 13  
NUMBER OF SEGMENTS: 19  
NUMBER OF NODES: 238  
NUMBER OF INTERFACES: 281  
NUMBER OF GATEWAYS: 21
```

This prints the summary information for the entire topology. If `ovtopodump` fails, this indicates some problem with `ovtopmd`. Try stopping all background services with `ovstop`, and then restarting them with `ovstart`.

If `ovstart` succeeds, but `ovtopodump` still fails, verify that at least one of the following is true:

- The IP address of the loopback interface is 127.0.0.1.
- There is an entry for `localhost` that has the correct address for the loopback interface.
- The IP address for the `localhost`'s hostname is correct. (That is, do an `ipconfig /all`, ping the hostname that was returned, and verify that the address returned is correct.)

On UNIX operating systems only, do a `hostname`, and then look up that name with `nslookup`, and verify that the address returned is correct.

If `ovtopodump` still fails, contact your HP support representative.

In addition, `ovtopmd` maintains consistency with the object database maintained by `ovwdb`. You can use the `ovtopofix` command to do the following consistency checks and updates:

- Check object existence in both the topology and `ovwdb` databases.
- Ensure that the managed state is consistent.
- Remove incomplete interface objects and incomplete nodes created by the discovery process from the object database.
- Update the presentation time to force `ipmap` to update the object's status during map synchronization.
- Update the creation time of the object to force `ipmap` to add the object during map synchronization.

To run `ovtopofix` and check for problems, type

```
ovtopofix -n
```

To run `ovtopofix` and fix any problems found, follow these steps:

1. Exit `ovw` using the `Map:Exit` menu item.
2. Stop `netmon` by typing **`ovstop netmon`**.
3. Type **`ovtopofix`**.



- Restart netmon by typing **ovstart netmon**.

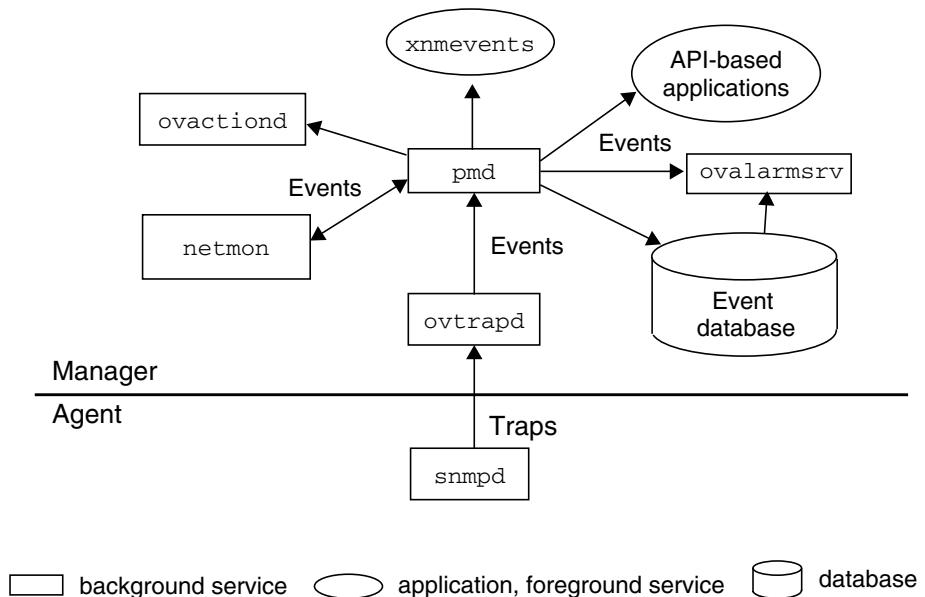
For more information, see the *ovtopofix* reference page in NNM's online help (or the UNIX manpage).

A license problem could also cause *ovtopmd* to fail. In this situation, verify that the license in *install\_dir\conf\.license* (*\$OV\_CONF/.license*) is correct.

## The ovtrapd Background Service

SNMP traps generated by an agent (*snmpd* in the case of the HP OpenView SNMP Agent) go to *ovtrapd* (via UDP port 162), which forwards them to *pmd*. *ovtrapd* acts as a buffer for traps that could potentially be dropped when *pmd* is busy. *pmd* logs and forwards the traps (in the form of an event) to *netmon*, *ovactiond*, *ovtopmd*, *ipmap*, and *xnmevents* (and potentially other subscribing applications). Figure B-2 illustrates this interaction.

**Figure B-2** **ovtrapd Event Interactions with SNMP Agent**



## The ovuispmd Background Service

You are most likely to see `ovuispmd` problems when you are trying to start `ovw`, since you cannot start `ovw` if `ovuispmd` is not running. You may also see `ovuispmd` problems when you use `ovpause` and `ovresume`, which are invoked when you perform a backup.

Problems that appear to be `ovuispmd` problems can actually be the result of:

- Problems with `ovuispmd` itself.
- Problems with one of the services on which `ovuispmd` depends.
- Problems with `ovw`, which depends on `ovuispmd`.

The easiest way to determine if the problem is with `ovuispmd` itself, or with another service, is to look at the symptom, as follows.

- When starting NNM, you get a dialog box saying that registration with the UI process manager failed. This is because `ovuispmd` is not running. `ovuispmd` must be running before `ovw` will run because `ovw` registers with `ovuispmd` when it initializes.
  - Use `ovstatus -v ovuispmd` to see if there is an informative exit status message for `ovuispmd`. Record the message and then try to restart `ovuispmd` using `ovstart -v` or `ovstart -v ovuispmd`. If the restart is successful, you should be able to start `ovw`. If you cannot start `ovw` after `ovuispmd` successfully restarts, then you probably have an `ovw` problem.
  - If `ovstart` fails and `ovuispmd` or any service that it depends on (`ovwdb` or `ovtopmd`) does not start, an error message will appear, indicating which service is not running. See the troubleshooting information for `ovspmd` or for the service that will not start.
- When you attempt to start NNM (or one of the `ovw` command line programs such as `ovw -mapcount`), `ovw` will not start.
  - Use `ovstatus -v` to determine if `ovuispmd` is paused. If it is, `ovw` will not start. This is expected behavior, to protect the NNM's paused state.

It should not take long for `ovpause` to complete. If the pause was initiated by the automated backup program, you can determine how long the system has been paused from the log file, `install_dir\tmp\ovbackup.log` (`$OV_TMP/ovbackup.log`).

If the paused state persists and there is no `ovbackup` in progress you can attempt to `ovresume` the system. Do not do this if a backup is in progress because it can result in corruption of the data store copies that are being created.

- You attempt an `ovpause` or `ovbackup.ovpl` when an `ovw` command line client is running, and `ovuispmc` denies the pause. This is expected behavior. Under these conditions the command `ovstatus ovuispmc` tells you that `ovuispmc`'s state is `PAUSE_ERROR` and `ovuispmc`'s last message field indicates that it denied the pause because a command line client is running.

To prevent database inconsistency or inaccurate reporting, `ovuispmc` denies a pause request if the pause request is received when one of `ovuispmc`'s registered `ovw` services is an `ovw` that was invoked with the `-mapcount`, `-deleteMap`, or `-copyMap` option.

— To resolve this problem, issue a new pause request after the `ovw` service terminates. These `ovw` command line services are designed to do specific, short tasks and then exit.

- There are `ovw` sessions that seem to be stuck pausing. If you have run `ovbackup.ovpl` and you find `ovw` sessions still paused after the background services have resumed, `ovuispmc` may have exited or been killed during the pause.

If a resume request is issued when `ovuispmc` is not running, then `ovw` sessions will not be notified of the resume and will remain paused, even when the background services have resumed. When `ovuispmc` terminates ungracefully it leaves a state file containing information about the registered `ovw` sessions. You can examine this file in `install_dir\conf\ovuispmc.state` (`$OV_CONF/ovuispmc.state`). *Do not modify this file!* It is used by `ovuispmc` when it is restarted.

— Use `ovstatus` to determine if `ovuispmc` is still running. If it is not running, restart it. When it initializes, it will automatically attempt to resume all of the `ovw` sessions that were registered with it when it terminated ungracefully.

- The `ovw` session information reported by `ovuispmc` is not correct. When you run `ovstatus -v ovuispmc`, it includes `ovw` sessions that are no longer running.

Under some termination conditions, `ovw` sessions are unable to unregister with `ovuispmc` before they terminate. In those cases, `ovuispmc`'s list of `ovw` sessions will not be corrected until `ovuispmc` attempts to communicate with the registered sessions.

- If no `ovw` sessions are running locally or remotely then it is safe to stop `ovuispmc` and restart it. Its `ovw` session information will be updated when it is restarted.
- `ovuispmc` exits ungracefully. This may be due to a termination signal from `ovspmd`.
  - See the *ovuispmc* reference page in NNM's online help (or the UNIX manpage) for information about forced termination. Also refer to the file `install_dir\lrf\ovuispmc.lrf` (`$OV_LRF/ovuispmc.lrf`) for useful comments about configuring the timeout field to prevent this occurrence.
- When you try to start `ovw`, `ovw` will not start or `ovuispmc` exits.
  - If `ovuispmc` is running but `ovw` does not start, or if `ovw` startup causes `ovuispmc` to abort, there may be useful information in the `ovuispmc` logging messages. The log messages may help you characterize the problem as an `ovw` problem, an `ovuispmc` problem, or an application problem.

Detailed instructions for using the Windows Application Event Log and the UNIX operating tracing and logging facilities, `nettl` and `netfmt`, are found in this appendix in the section “Logging and Tracing” on page 602. Examine the logging output for `ovuispmc` errors in the `OVEXTERNAL` subsystem section. Also refer to the diagnostics in the *ovuispmc* reference page in NNM's online help (or the UNIX manpage).

## The `ovwdb` Background Service

The `ovwdb` service controls the object database. Here are some problems that can be caused by `ovwdb` and possible solutions to the problems.

`ovwdb` should be running whenever NNM is running. Normally it is started through `ovstart`. To verify that it is up and running, type:

```
ovstatus ovwdb
```

You should see something like this:

```
object manager name:  ovwdb
state:                RUNNING
PID:                 4115
last message:        Initialization complete.
exit status:         -
```

By default, `ovwdb` uses a TCP socket bound to port 2447. If `ovwdb` indicates an error attempting to bind to this port number because another application is already using it, follow these steps:

1. Make sure that you do not have another `ovwdb` service using port 2447. Check the `/etc/services` file for a listing of ports being used.

The most common reason for errors binding to the TCP socket is the presence of another `ovwdb` service. If an `ovwdb` service was started outside of `ovstart`, the `ovstop`, `ovstart`, and `ovstatus` commands will not recognize this service. To prevent this problem, always start all NNM background services with the `ovstart` command and stop them with the `ovstop` command.

If another application is using port 2447, continue with the next step.

2. Change the port number to any unused port by changing the entry for `ovwdb` in `%SystemRoot%\system 32\drivers\etc\services (/etc/services)`.

```
ovwdb 2447/tcp # OpenView Object Database daemon
```

3. Reconfigure NNM by typing the following commands:

```
ovstop
```

```
ovstart
```

4. If `ovwdb` still fails to run successfully from `ovstart`, contact your HP support representative.

---

**NOTE**

If you are using the Network Information Service (NIS), make sure you update your NIS Master Server.

---

If you notice a great deal of swapping and the system performance does not meet your expectations, you can do one of two things:

- Edit the `install_dir\lrf\ovwdb.lrf` (`$OV_LRF/ovwdb.lrf`) file to configure the maximum number of objects that will be in the `ovwdb` cache.
- Change the system parameters (kernel parameters and regenerate the kernel). For more information, see “Improving Traffic Management and Performance” on page 617 in this appendix.

To verify the status of `ovwdb`, first run the `ovstatus -v` command to determine if everything is working correctly. If `ovwdb` stops running, `netmon` and `ovtopmd` will also stop running because these services are dependent on `ovwdb`. On UNIX operating systems only, if `ovwdb` is no longer running, check the `nettl` log using `netfmt`. Note that this assumes `nettl` was running when the problem occurred. To verify that `nettl` is running, execute the `nettl -status` command. See “Using `nettl` and `netfmt` (UNIX operating systems only)” on page 606 for more information on using `nettl`.

If you exhaust all available memory, the `ovwdb` service will stop running. To solve this problem, follow the suggestions described above for solving problems with excessive swapping; that is, either edit the `ovwdb.lrf` file or change the system (kernel) parameters.

If `ovwdb` stopped running due to memory problems, a disaster message will be logged into the Application Event Log (through `nettl`).

If the object database gets corrupted in any way, for example, by removing the files in the `install_dir\databases\ovwdb` (`$OV_DB/ovwdb`) directory, you need to restart the automatic map generation from the beginning.

## The `pmd` Background Service

The `pmd` service is the postmaster background service. It provides event (trap) forwarding and message passing between services. When `pmd` receives an event, it forwards it to all the services that are registered to receive the event. Events are generated by these services:

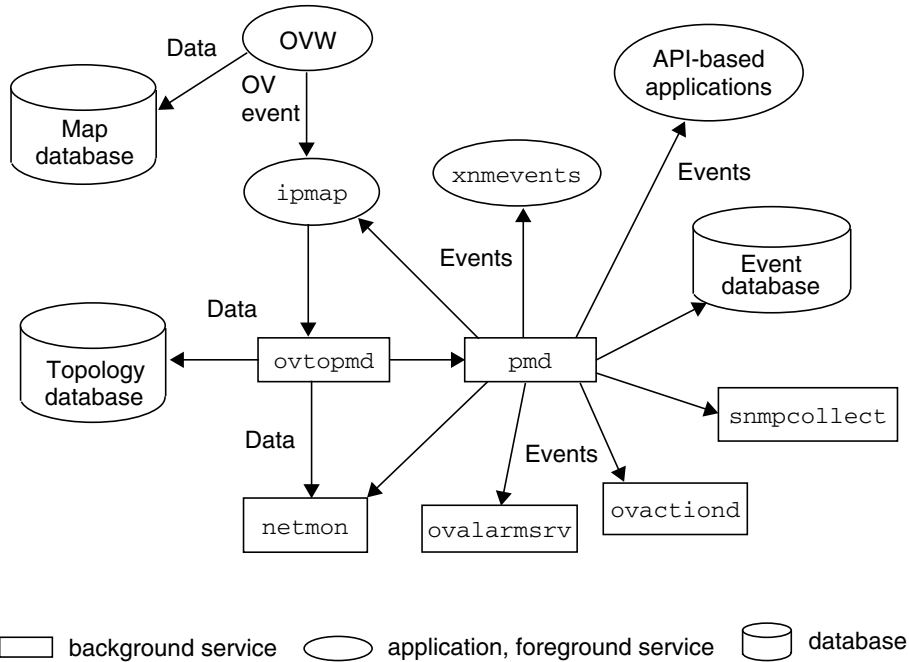
- `ipmap` and `ovtopmd` when you edit the map or when nodes’ status propagates to segments and networks.
- `netmon`, when it detects a change as a result of polling nodes or as a result of receiving a trap forwarded by `pmd`.
- `ovtrapd`, when `ovtrapd` receives a trap from agent services.

- `ovtopmd`, when topology changes occur.
- `snmpCollect`, when a threshold is exceeded or reset.
- `ovevent`, when this command is called from a batch file (shell script).
- Configuration operations such as `xnmloadmib`, `xnmtrap`, `xnmsnmpconf`, and `xnmcollect` when you edit configuration files.
- `ovrepld` as a result of a change detected on a remote collection station. Refer to *A Guide to Scalability and Distribution* for information on the `ovrepld` background service and collection stations.
- `ovspmd`, during pause and resume operations.

Figure B-3 illustrates the interactions between services and `pmd` when you add an object to a map in NNM. To notify NNM about the new object, `ovw` sends an `ovw` message to `ipmap`. `ipmap` tells `ovtopmd` to update the topology database. `ovtopmd` responds by updating the topology database and sends a new node event to `pmd`. `pmd` logs the event in the event database and forwards the new node event to `netmon`, `snmpcollect`, and `xnmevents` (and potentially other subscribing applications). When `netmon` receives the event, `netmon` pulls the information it needs from

the topology database through `ovtopmd`. When `snmpCollect` receives the event, it checks to see if the new node should be collected from (that is, whether it matches any configured IP wildcards).

**Figure B-3** **pmd Event Interactions**



For a discussion of `pmd`'s interactions and operation concerning events generated by `netmon`, see “The `netmon` Background Service” on page 548.

If you suspect a problem with the `pmd` background service, try the following actions to further isolate the problem:

- If `pmd` is not configured to write to `eventdb`, you'll first need to execute `ovdumpevents` to write the contents of the event database to a log file.
- Look for events logged to the event database, using `xnmevents` or the Alarm Browser.



- Add a node to the map and then check to see if the event was recorded in the event database, using `xnmevents` on a management station or the Alarm Browser through the Web UI.
- Run `ovstatus` and check the status of `pmd`. If the `pmd` service appears to be hung, stop the service using `ovstop` and restart all background services with `ovstart`.
- Check `pmd.log0` to see any messages related to the `pmd` service.
- Turn on tracing as described in the “Recommended Logging and Tracing Practices” on page 602 in this chapter. Your HP support representative can help you interpret the trace results.
- Verify that events can be generated. Use the `popupMsg` contributed script in `$OV_CONTRIB/NNM/popupMsg` to generate an event.

## The `ovrequestd` Background Service

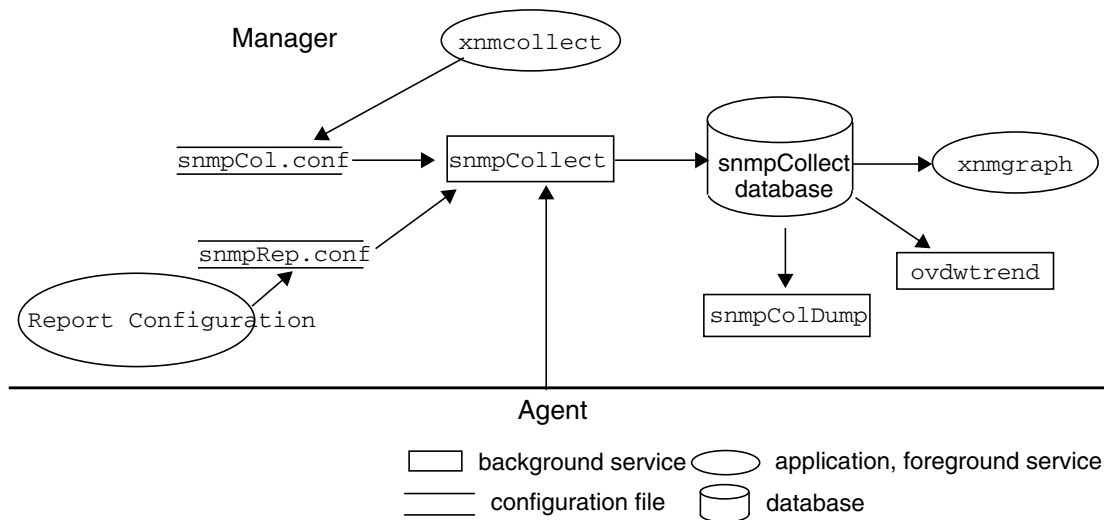
The `ovrequestd` service executes reports and data warehouse exports according to a predefined schedule. Once a report is configured, `ovrequestd` starts executing the exports and the reports.

Refer to the *Reporting and Data Analysis* online book for more information.

## Troubleshooting Data Collector Problems

Figure B-4 shows the interactions among the services `xnmcollect`, `xnmgraph`, `snmpCollect`, `snmpColDump`, and an SNMP agent.

**Figure B-4** `snmpCollect` Interactions with SNMP Agents



If you have problems collecting MIB data, check for these problems:

- The service for the data collector, `snmpCollect`, may not be running on the management station.
  - Make sure that you are not out of disk space. The data collector stops running when the disk is full. See page 572 for information about what to do if the disk is full.
  - Verify the `snmpCollect` service is running by executing the `ovstatus snmpCollect` command. If the `snmpCollect` service is not running, restart the data collector using the `ovstart` command.
- Data is not being collected from the node it should be or thresholds are not being generated.
  - Make sure the target node supports SNMP.

- Make sure the node is managed, or that the `-u` option to `snmpCollect` is used.
- Verify that the MIB object is configured for collection. Use the `Options>Data Collection & Thresholds:SNMP` menu item to verify the configuration.
- Verify that the target node is configured to collect data or generate threshold events. You can configure the node to collect data by explicitly specifying the node name, by specifying a wildcard, or by specifying the name of an ASCII file that contains the node name. Also, verify that the target node has not been excluded for collection. Use the `Options>Data Collection & Thresholds:SNMP` menu item to verify the configuration.
- Ensure that collections for the MIB object are not in Suspended state.
- Ensure that the desired instances are specified for the MIB object and node combination.
- Perform `Fault:Test IP/TCP/SNMP` on the collection in question. This checks the connection between the local management station and other SNMP nodes.
- If `sysObjectID` filtering is being used, make sure the correct `sysObjectID` is being used.
- See if other collections or thresholds from that node are working. If none are working, the problem is probably with the node. From the `Options>Data Collection & Thresholds:SNMP` window, select the MIB object, and then select `Actions:Show Data` to see if data is being collected from the node.
- Try to access the MIB object using the `Tools:SNMP MIB Browser` dialog box.
- Run `snmpCollect -S` and look at the `install_dir\log\($OV_LOG)snmpCol.trace` file. This command lists the configuration of the data collection.
- Try turning on verbose tracing. Execute `ovstop snmpCollect`. Then execute `ovstart snmpcollect -v`. Check the output in the `snmpCol.trace` file.

- If you expect threshold events to occur, but they do not, start a data collection on the MIB variable. The problem may be that the threshold value is not set correctly. For example, if the normal value for this variable is 20 and you set the threshold value to 100, you will probably never see a threshold event.
- If you can't resolve the problem, use the tracing command. To use the tracing command, try restarting the data collector with the same command as in the *install\_dir\lrf\snmpCollect.lrf* (*\$OV\_LRF/snmpCollect.lrf*) file, but add the *-t filename* option. The *-t* option is a tracing option that logs the results to the *filename*. Run `snmpCollect -T` to toggle the tracing. To determine whether or not tracing is on, look at the trace file. See the *snmpCollect* reference page in NNM's online help (or the UNIX manpage) for more information.
- For more information about troubleshooting threshold events, see the section "Configuring Events" on page 594 in this appendix.
- The disk is full. The data collector stops running when the disk is full. To solve this problem and to prevent this problem from happening again, do the following:
  - Use `ovdwtrend -delpriorto date/time`. This command will export `snmpCollect` data to the data warehouse and then delete the `snmpCollect` data after it has been exported. See the *ovdwtrend* reference page in NNM's online help (or the UNIX manpage) for more information on using this command.
  - Free up disk space by removing files in the *install\_dir\databases\snmpCollect* (*\$OV\_DB/snmpCollect*) directory. Note, however, when you remove files you lose the data you have saved. If you don't want to remove the whole file, you can edit the file and keep only the most recent data by deleting older entries in the file. Once you have freed up disk space, as root restart the data collector with the `ovstart` command.
  - Edit files in the *install\_dir\databases\snmpCollect* (*\$OV\_DB/snmpCollect*) directory using the `snmpColDump` command. For example, to retain the last 100 entries of a file, use the following commands (available in the MKS Toolkit on a Windows operating system):

```
snmpColDump -tTI snmpCollect\filename\
awk -F\t '{printf("%d\t%d\t%s\t%1g\n", $4, $5, $6, $3)}' | \
tail -100 > \tmp\save
snmpColDump -r \tmp\save install_dir\databases\snmpCollect/
```

**On UNIX operating systems only:**

```
snmpColDump -tTI $OV_DB/databases/snmpCollect/filename\
awk -F\t '{printf("%d\t%d\t%s\t%1g\n", $4, $5, $6, $3)}' | \
tail -100 > /tmp/save
snmpColDump -r /tmp/save $OV_DB/databases/snmpCollect/
```

- If you want to stop all data collection, kill the `snmpCollect` service with the `ovstop snmpCollect` command, or suspend all data collections using the Options>Data Collection & Thresholds:SNMP menu item.

To prevent the problem in the future:

- Lengthen the polling intervals. Using the Options>Data Collection & Thresholds:SNMP menu item and set the polling interval to not collect as often. In addition, if you only want to check thresholds, don't store the data.
- Reduce the number of MIB objects you are collecting on.
- Reduce the number of nodes you are collecting on.
- Reduce the number of instances you are collecting for some or all nodes.
- Set up programs (`cron`) to periodically either remove files in the `install_dir\databases\snmpCollect` (`$OV_DB/snmpCollect`) directory or export and trim them with `ovdwtrend`.
- There is no response when querying DMI devices.
  - Make sure that DMI is configured correctly on the remote device.
  - The Data Collector supports only SNMP devices. Install an SNMP agent and a DMI-to-SNMP mapping agent on the DMI system.
- The data you are collecting appears to be incorrect.

If you suspect that a simple variable collection is inaccurate, check the values reported by the agent using one of the following methods:

- Use `snmp` commands such as `snmpwalk` and `snmpget`.

- Use NNM's mib browser from the Tools: SNMP MIB Browser menu.
- Open the Data Collections menu and use the Actions: Test SNMP command.
- Turn on `snmpCollect` tracing. See the `snmpCollect` reference page (or the UNIX manpage) for details.

If you use MIB expressions, take each of the collected values and manually calculate the expression. This is easier if you use `snmpCollect` tracing and enable the SNMP packet tracing with the `-d` option.

If you collect data that provides a rate of change calculation, check the collected values and the `sysUptime` MIB variable that provides the time delta. This time delta should be in hundredths of a second. Once again `snmpCollect` packet tracing is helpful.

## Set-community Name and Trap-dest Loss Upon Re-Installation

The `emanate` SNMP agent uses the following two configuration files in the `install_dir\conf\SnmpAgent` (`$OV_CONF/SnmpAgent`) directory:

- `snmpd.cnf` — the SNMPR-format `emanate` configuration file.
- `snmpd.conf` — the HP-format SNMPD configuration file.

`emanate` uses the union of configuration information from both files. However, some of the information is in both—`sysDescr`, `sysName`, `sysContact`, and `sysLocation`. When `emanate` starts up, it uses the `snmpd.conf` values (if defined) and copies them to `snmpd.cnf`; otherwise, it uses the values in `snmpd.cnf`.

The `get-community`, `set-community`, and `trap-dest` fields in `snmpd.conf` are not replicated in `snmpd.cnf`, so they are appended (in `emanate` run-time memory) to the set of configurations in `snmpd.cnf` because it is allowable to have more than one value for each.

Thus, upon installation, the default `snmpd.cnf` file is conditionally installed by `Setup` (only if one doesn't already exist)—but the default `snmpd.conf` file is always installed. After uninstallation and re-installation, the result is that `sysContact` and `sysLocation` get

taken from the existing `snmpd.conf` file, but the `set-community` and `trap-dest` are taken from the new/default `snmpd.conf` file (in which they are not defined).

The solution is to conditionally install `snmpd.conf` only when it does not already exist.

## Troubleshooting Web Components

If you experience problems starting any of the web components, you should first check the following before doing further troubleshooting.

- Make sure that NNM is installed on the host specified in the URL, and that the host is running.
- Make sure that you are using a supported web server, and that it is configured properly and is running. Supported web servers include Microsoft Peer Web Services, Microsoft Internet Information Server, and Apache (shipped with NNM for UNIX operating systems.) Try entering the simplest URL (`http://hostname[:port]`) in the web browser and see if the server is accepting connections.
- Make sure that you are using a web browser that supports NNM. Supported web browsers include Netscape 4.6 or greater and Microsoft Internet Explorer 5.0 or greater.
- If starting the web component directly from a URL, make sure the full URL was entered correctly. If NNM is running on a UNIX operating system, check that port 3443 is specified in the URL.
- Make sure that the hostname in the URL is fully qualified. Do not use the short version of the hostname. For example, use `mssystem.home.com` instead of just `mssystem`.
- Exit and restart the browser, and then re-enter the URL.
- Clear the memory and disc caches for the browser, and restart the browser.

In Netscape Navigator, select the Edit:Preferences menu item. In the dialog box, click on Advanced, then click on Cache. In the next dialog box, click on Clear Disk Cache and Clear memory Cache.

In Internet Explorer, select the View:Internet Options menu item. On the General page, select the Temporary Internet Files field. Then click on the Delete Files button.



- Check the contents of all of the web log files for possible errors.

*Windows:*

```
install_dir\www\logs\launcher\error_log  
install_dir\www\logs\launcher\login_log  
install_dir\www\logs\launcher\access_log
```

*UNIX:*

```
/var/opt/OV/www/logs/launcher/error_log  
/var/opt/OV/www/logs/launcher/login_log  
/var/opt/OV/www/logs/launcher/access_log
```

- If an http proxy server is in use, try disabling the web browser from using it.

In Netscape, select the Edit:Preferences menu item. In the dialog box, click on Advanced, then click on Proxies. In the dialog box, select Direct connection to the Internet.

In Internet Explorer, from the General page select the View:Internet Options menu item. On the Connection page, deselect Access the Internet using a proxy server.

- For errors in the Reporting interface, refer to the *Reporting and Data Analysis* online manual.

## Language Selection Problems

NNM's web applications are available in English and Japanese. However, it is possible to integrate and use other web applications that have been translated into other languages while NNM's web applications remain in English. If you expect the web applications to be in a language other than English and they are not, it could be due to one of the following problems:

- If the strings are appearing in English, check the browser's language configuration.

In Netscape, Languages is configured through the Netscape:Preferences menu item.

In Internet Explorer, Languages is configured through the Internet:Options menu item.

In both cases, the language that you expect to see should be listed at the top of the `Languages` window. To see NNM's web applications in Japanese, be sure that the Japanese entry (`ja`) is listed before the English entry (`en` or `en-us`). Restart the browser after making changes.

- If the non-English strings appear garbled in the window, check the following:
  - The browser's encoding or font configuration may be incorrect.

In Netscape, the encoding is specified through the `View:Encoding` menu item.

In Internet Explorer, the font is specified through the `View:Font` menu item.

To choose Japanese, one of the Japanese fonts or encodings should be selected.
  - On UNIX systems only, the browser was not started with the correct environment variable settings.

The browser process should be started with the correct `$LANG` setting. For Japanese, `XMODIFIERS` must also be set. These variables are usually set by the desktop environment.
  - On Japanese on UNIX systems only, the default codeset in use by the server may be incompatible with the Web applications.

By default, NNM's web applications expect the data from the servers to be in Shift-JIS codeset for HP-UX and Windows operating systems. On Solaris and Linux, EUC is the default. Use the `ovchange_locale_mapping` command to switch the default configuration. See the `ovchange_locale_mapping` manpage for details.

## HP OpenView Launcher

### User Login Problems

If you expected to get a user login screen when the Launcher starts, but you did not, then the file `session.conf` probably has not been set up for user logins. See Chapter 14, *NNM on the Web*, for information on setting up users and logins.

If you get the user login screen and the Launcher does not accept the login, check the following:

- Make sure you typed the user name and password correctly.
- Make sure the password file, `htpasswd`, includes the user name and password. See Chapter 14, *NNM on the Web*, for information on setting up the `htpasswd` file.
- If running NNM in a non-English language, be sure that the environment variable `XMODIFIERS` is set correctly.

### Starting the Launcher

If you have problems starting the Launcher, check the error log file for potential errors. The Launcher error log records internal errors from the Launcher. You can use the output from this file when discussing a problem with the HP Response Center or support personnel.

If the error log file does not give you sufficient information for troubleshooting, you can set a parameter in the URL for `ovlaunch.exe` that will enable more in-depth error logging on a session level.

The parameter is:

`http://hostname:[port]/OvCgi/ovlaunch.exe?Debug=/tmp/file`

The output of this file is less “friendly” than other error log files, but it contains valuable information that may be useful to a support person who is trying to assess the problem.

If the Launcher does not start and you get an error message indicating that there is a CGI problem, then the problem could be with one of the Launcher’s CGI programs. Try executing each of the following programs from the command line. These programs are located in:

*Windows:* `install_dir\www\cgi-bin`

*UNIX:* `/opt/OV/www/cgi-bin`

- `ovlaunch.exe`
- `ovlogin.exe`
- `ovsessioninfo.exe`
- `ovlaunchreg.exe`

You can tell if these programs are running if they output ASCII text to the screen. If one of these programs is not running, contact HP Support.

## Launcher Window Problems

If the Launcher window comes up with no content (blank), it could be one of the following problems:

- The correct patches have not been installed on the system. Check the system requirements section of the Installation Guide.
- You could have a conflict if you had an older version of Netscape installed on your system. If you have set the `CLASSPATH` environment variable when running an older version of Netscape, it will cause Java access problems when you run the recommended version of Netscape. If you have set this variable, you should unset it before running the newer version of Netscape.
- You could have a Java applet problem. Check the Java log as follows:
  - In your web browser, check to see that the items Java and Java script are enabled. In Netscape Navigator, check this by using `Edit:Preferences->Advanced`. In Internet Explorer, check this by using `View:Advanced`.
  - Bring up a Java console and view the messages it contains. In Netscape Navigator, use `Window:Java Console`. In Internet Explorer, use `View:Java Console`.
  - Try starting the Launcher without going through a proxy.

In Internet Explorer, select the `View:Internet Options` menu item. Choose the `Connection` tab, then turn off the proxy server.

In Netscape Navigator, select the `Edit:Preferences` menu item, choose `Advanced`, and disable the proxy server.
  - The Java console also provides a method of logging information about the Java virtual machine interpretation of the Java code that is executing. Both of the supported web browsers allow you to turn on Java error logging from the browser menu. In Netscape Navigator, look under the `Window` menu item. In Internet Explorer, select the `View:Internet Options` menu item under the `Advanced` tab.

If you encounter errors when reading in registration files, run the `regverify` command in `install_dir\bin\` (`$OV_BIN`) against the Launcher Registration File (WLRF) in the following directory:

*Windows:* `install_dir\www\registration\launcher\%LANG%`

*UNIX:* /etc/opt/OV/share/www/registration/launcher/\$LANG

You can also hold down the **Shift** key and press the [Reload] button on your browser to force the registration files to reload.

### Other Sources of Information

Web server log files are available that provide you with information about server-related problems. The error log is located at:

*Windows:* \WINNT\System32\Logfiles

*UNIX:* /opt/OV/httpd/logs

This is the default location, but it may vary depending on where your web service is set to log errors. Check your web server properties to verify the location.

## Network Presenter

If the Network Presenter does not start, check the items on page 576, and the following:

- If the Network Presenter progress bar does not appear, it could indicate web server problems. Try connecting to the web server from your browser without starting Network Presenter. For example:

**http://hostname:3443**

- Make sure the Network Presenter is registered with NNM. In NNM on a management station, use the Help>About HP OpenView menu. In the dialog box, click Applications. Verify that the Network Presenter appears in the list. Alternatively, check the registration directory to ensure that the file netpresenter appears. The location is:

*Windows:* install\_dir\share\registration\%LANG%

*UNIX:* /etc/opt/OV/share/registration/\$LANG

- Make sure that NNM is running on the management station, with the map opened to which the Network Presenter is trying to connect.
- If you tried starting the Network Presenter from the Tools or the Objects selection in the Launcher, try starting it directly from the URL instead.

If there are missing menu bar or toolbar items, do the following:

- Check the permissions on the registration files registered with Network Presenter. The permissions need to be read by all, and the owner and the group should be set to bin. The registration files are located in:

*Windows:* `install_dir\www\registration\jovw\%LANG%`

*UNIX:* `/etc/opt/OV/share/www/registration/jovw/$LANG`

- Set the `QUERY_STRING` environment variable to `menus`. Then run `jovwreg.exe` from the command line on the system running NNM. Alternatively, from the browser, use the URL  
`http://hostname:[port]/OvCgi/jovwreg.exe?menus`
- Run the `regverify` command `install_dir\bin\ ($OV_BIN)` against the Network Presenter Registration Files (NPRF).

If symbols are missing from the Network Presenter, do the following.

- Set the `QUERY_STRING` environment variable to `symbols`. Then run `jovwreg.exe` from the command line on the system running NNM. Alternatively, from the browser, use the URL  
`http://hostname:[port]/OvCgi/jovwreg.exe?symbols`

If the Network Presenter connected to an `ovw` session that you think you should not be connected to:

- In the Network Presenter, bring up the `Help>About HP OpenView` dialog box. Verify that the session ID is the session ID in the `ovw` session to which you wanted to connect.

If you are not getting dynamic updates to your map:

- Verify that you are connected to a read/write version of the map. To do this, compare session IDs in the Network Presenter and in NNM on the management station. If both a read/write and read-only version of a map are running on the same system, and you want to connect to the read/write version, ensure that the read/write version of the map is started in NNM before the read-only version.

## Alarm Browser

If the Alarm Browser does not start when requested, check the items on page 576, and the following:

- Run `ovstatus ovalarmsrv` on the command line. This will tell you if the `ovalarmsrv` service is running. If it is not, execute `ovstart ovalarmsrv`.
- If `ovalarmsrv` is running, stop it using `ovstop ovalarmsrv`, then restart it using `ovstart ovalarmsrv`. This usually clears up the problem.
- `ovalarmsrv` reads from the event database. Make sure that there is not a problem with this database. Make sure that the `eventdb` directory in `install_dir\databases\($OV_DB)` and all the files in that directory have read permissions.
- `ovalarmsrv` depends on the `pmd` service to be running. Check to be sure that `pmd` is running.
- Make sure that the specification file `install_dir\www\conf\NNM.spec` (`$OV_WWW/NNM.spec`) is present and that it has read/write permissions.

If you have trouble running the Alarm Browser:

- Clear the cache in your browser. See the procedure on page 578.
- Use the options to the `ovalarmsrv` command to change the parameters to the command. See the *ovalarmsrv* reference page in NNM's online help (or the UNIX manpage) for more information.

## SNMP Data Presenter

If the SNMP Data Presenter does not start, check the items on page 576, and the following.

### SNMP Data Presenter is Empty

If the SNMP Data Presenter does not show any data, check the following:

- Make sure JavaScript is enabled in your browser. In Netscape Navigator, select `Edit:Preferences->Advanced`. In Internet Explorer, Java is always enabled.
- Make sure `ovsessionmgr` is running on the server. To check:  
*Windows:* Use the Task Manager.  
*UNIX:* Type `ps -ef | grep ovsessionmgr`

If it is not running, then start it by entering the URL for `ovlaunch.exe`.

### **Access Denied**

If you receive an error message indicating that access is denied, that a violation occurred, or that data was not found, check the following:

- Check your web server configuration. Verify the port number, alias names, directory names, and permissions. For more information, consult your web server documentation.

If you receive a message that you have a CGI Error or an Internal Server Error, please contact HP Support.

### **If No Data or Wrong Tree Structure Appears**

If no data or the wrong tree structure is shown on the scoping pane:

- Verify the application registration files (ARFs) on the management station. Type:

*Windows:* `install_dir\bin\regverify`

*UNIX:* `$OV_BIN/regverify`

(Set the `LANG` environment if necessary.)

Alternatively, you can enter

`http://hostname:[port]/OvCgi/snmpviewer.exe?ins=check` in your browser.

If there are errors, fix the errors by editing the ARFs.

### **New Applications Not Displayed**

If you added new applications with Application Builder, but they are not displayed on the scoping pane, clear the cache on your browser following the instructions on page 578, then launch SNMP Data Presenter again.

### **Results Not Shown, or are Incorrect**

If the scoping pane is blank after resizing the browser window or frames:

- Click with the right mouse button on the scoping pane and select `Frame Reload`.



If you click an item in the scoping pane, but the results are not shown in the results pane:

- Check if the target nodes are healthy. Try ping or SNMP commands on the management station to the target nodes. See “Network Management Operations” on page 589 for more information.

If results are shown, but errors are indicated:

- All the operations in SNMP Data Presenter are defined in ARFs on the management station. NNM on the management station has the same menu items as in SNMP Data Presenter. Try the operation on the management station. If you still encounter errors, see the sections “Network Management Operations” on page 589 and “Executing MIB Applications” on page 592.

If the format in the results pane is unappealing:

- Make sure a fixed font is specified in the fixed width font setting. In Internet Explorer, select the View:Internet Options->Fonts menu item to change fonts. In Netscape Navigator, select the Edit:Preferences->Appearance->Fonts menu item.

## Troubleshooting NNM Operations

This section suggests actions to take if you suspect a problem with one of the operations of NNM. This section also discusses the sequence of interactions and/or the flow of data associated with an operation when such information may help you to isolate the problem. This section covers the following:

- Runtime components.
- Network management operations.
- Browsing MIBs.
- Building MIB applications and executing MIB applications built by the MIB Application Builder.
- Collecting MIB information from network nodes at regular intervals.
- Configuring events.
- Event Reduction Capabilities
- Loading MIBs.
- X Windows components.
- Online help.

For detailed information on a command or service mentioned in this section, see its associated reference page in NNM's online help (or the UNIX manpage).

### Runtime Components

If you are having problems running NNM, you should check file permissions, start-up scripts, and error log. These topics are discussed below.

#### Manager File Permissions

By default, the `ovw` (graphical network map) program is executable by anyone; the `ovwdb`, `pmd`, `snmpCollect`, `ovactiond`, `ovtopmd`, and `netmon` background services are executable only by `root`.

The remainder of this section is for UNIX operating systems only.

You may restrict the permissions on the map files using the `$OV_BIN/ovwchmod`, `$OV_BIN/ovwchgrp`, and `$OV_BIN/ovwchown` commands. These commands, along with the `ovwls` command, are explained in the *ovwperms* reference page in NNM's online help (or the UNIX manpage). These commands change all of the map data files associated with a map. You use the commands the same way as the operating system `chmod`, `chgrp`, and `chown` commands, except that you follow them with a map name instead of a file name. Note that these commands do not change permissions on the product's directories or executable files, nor on other configuration and data files.

---

**NOTE**

---

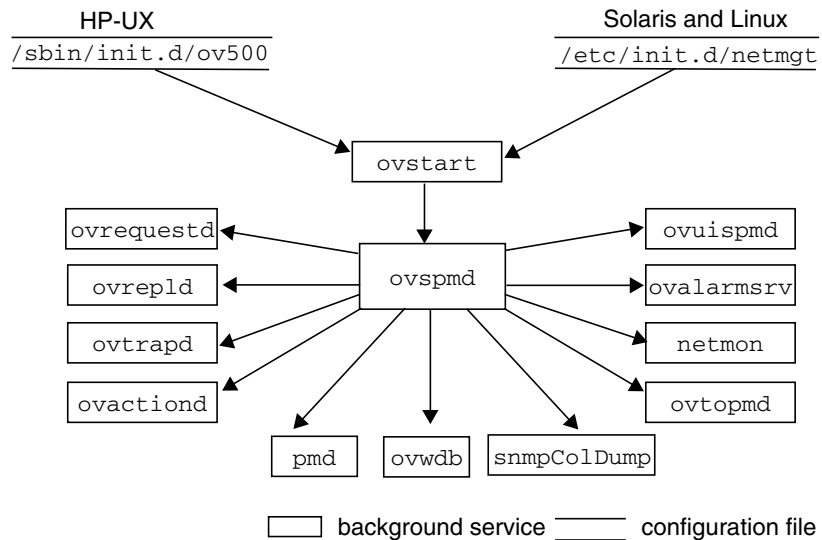
Do not change map permissions when NNM is running.

If you are having problems running NNM, try restoring the map file permissions to their default using the `ovwchmod` command with mode `0777`.

### Start-Up Scripts (UNIX operating systems only)

Check to see if some component in the execution sequence is “broken,” such as a syntax error in one of the product’s start-up scripts. Figure B-5 shows the start-up execution sequence for NNM. All files are in the \$OV\_BIN directory.

**Figure B-5** Manager Startup



A script file automatically executes `ovstart` when you reboot the system. The script file has the following name:

- *HP-UX*: `/sbin/init.d/ov500`
- *Solaris and Linux*: `/etc/init.d/netmgt`

If `ovstart` does not execute, the problem may be with the start-up script. The script file is created during installation. If the file already exists, the existing file is moved to another file, and the new one is copied to the original file name. For example, on HP-UX systems the existing file would be moved to `/sbin/init.d/#ov500`, and the new file would be copied to `/sbin/init.d/ov500`.

## Network Management Operations

If you are having problems running an operation to a remote node, check the following items.

- If you configured agent community names, verify that the node's community name listed in the `Options:SNMP Configuration` dialog box matches the community name configured on that node.
- The `Locate Route` operation could fail if one of the nodes in the route uses a routing algorithm that is not based on the BSD (Berkeley Software Distribution) 4.2 routing algorithm or does not support SNMP.

The `Locate Route` menu item could also fail if all IP addresses for a gateway do not resolve to the same IP hostname.

- Ensure that the remote node supports the operation. Most of the NNM operations require SNMP-based, MIB-I (RFC 1156) or MIB-II (RFC 1158) compliant agent software on the remote node.
- On UNIX operating systems only, if you modified any X resources for default network management operation options, ensure that the modifications are correct. You may want to restore the default X resources as explained in “X Windows Components (UNIX operating systems only)” on page 598 in this appendix.

### Execution Models for Network Management Operations

Some of the NNM operations start directly from `ovw`; others are started by an integrating service, `xnmappmon`.

Figure B-6 shows some of the network management operations started (or startable) directly by `ovw`.

**Figure B-6 Applications Started Directly by the `ovw` Service**

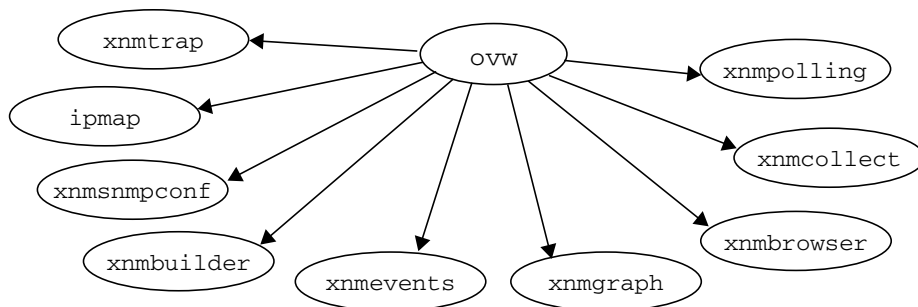
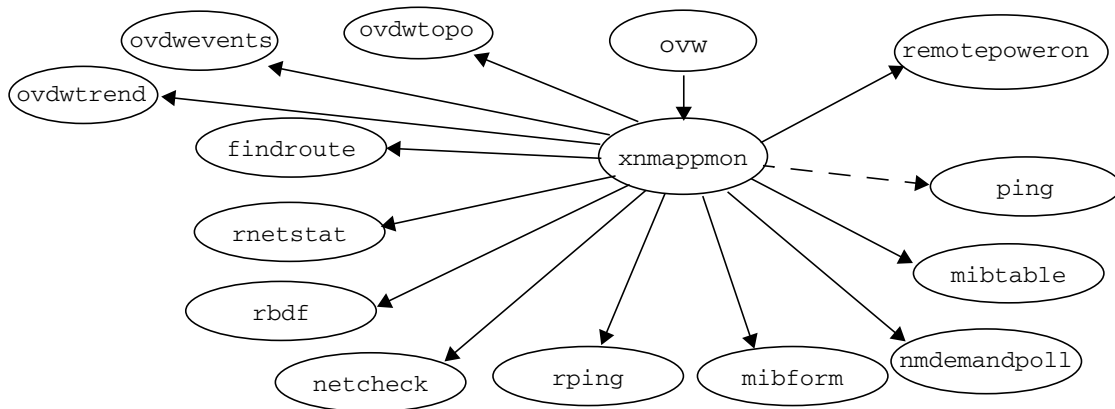


Figure B-7 shows those operations started by `xnmappmon`.

**Figure B-7 Applications Started by `xnmappmon`**



Many of the services shown in the previous two figures may be run as commands for remote troubleshooting. For information about which commands are executed for the different services, see each service's reference page in NNM's online help (or the UNIX manpage). `ping` is shown as a dotted line because it is a system command.

## Managing DMI Devices

NNM supports SNMP communications. In order to access DMI devices, you must install a DMI-to-SNMP mapping agent on the DMI system. Then SNMP requests will get processed.

When using the DMI Browser, keep in mind that it only supports DMI communications with HP-RDMI version 1.1 service providers. Use `Edit:Find->Object By Attribute` and search on:

- The DMI Version attribute for the 1.1 value.
- The DMI RPC attribute for the HP-RDMI value.

## Browsing an Internet MIB

If you have problems browsing an Internet MIB using the `Tools:SNMP MIB Browser` menu item, check the following:

- Make sure the Internet MIB is loaded. Use the `Options:Load/Unload MIBs:SNMP` menu item to see what modules are currently loaded. If the MIB is not listed, use the `Load/Unload MIBs` menu item to load the Internet MIB into the MIB description file.
- Make sure that the version of the Internet MIB loaded into NNM matches the Internet MIB version that is implemented on the device being managed. When you get a new device on your network or upgrade an existing one, make sure that you get the Internet MIB for that device.

If you are unable to set the value on an object so you can physically update the device, make sure that the agent is configured to respond to SNMP `set` requests. To do so, enter a community name that will allow `set` requests on the node using the `Options:SNMP Configuration` menu item, or enter the community name within the `Browse MIB` dialog box.

## Building and Executing MIB Applications

The problems you may encounter while building MIB applications and executing MIB applications built by the MIB Application Builder are closely related.

For example, some errors made while building an application may not show up until you actually execute the application. If the problem is an incorrectly built application, you need to modify your application.

This section is organized as follows:

- Things to check when you have problems building an application.
- Things to check when you have problems executing an application. Problems executing applications can be caused by either an error in the way the application was built, or by independent things such as the community name not being configured correctly.

### Building MIB Applications

If you have problems building MIB applications using the `Options:MIB Application Builder:SNMP` menu item, check the following:

- Make sure to read the instructions in online help about adding MIB applications.
- Make sure you have the enterprise-specific MIB loaded. Use the `Options:Load/Unload MIBs:SNMP` menu item to see what modules are currently defined. If the MIB is not listed, use the `Load/Unload MIBs` menu item to include the MIB in the loaded MIB database.
- Make sure the MIB object ID you select is supported by the device you are trying to monitor. To verify this, use the `Tools:SNMP MIB Browser` menu item.
- When you create a Table application, you can only select fields from a single table in the MIB tree.
- Make sure that the version of the MIB loaded into NNM matches the MIB version that is implemented on the device being managed. When you get a new device on your network or upgrade an existing one, make sure that you get the MIB for that device.

### Executing MIB Applications

If you have problems executing applications built using the `Options:MIB Application Builder:SNMP` menu item, check the following:

- Make sure you have the enterprise-specific MIB loaded. The application does not come up when the MIB for that particular application is not loaded. Use the `Options:Load/Unload MIBs:SNMP`



menu item to see what modules are currently defined. If the MIB is not listed, use the `Load/Unload MIBs` menu item to load the MIB into the MIB description file.

The MIB may be missing; for example, if you load a MIB, build an application, and then unload the MIB or if you copy the application to a new machine and the MIB is not on that machine. Make sure that if you copy the MIB application to another system, you also load the enterprise-specific MIB on that system.

- Make sure the MIB object ID you select is supported by the device you are trying to monitor. To verify this, use the `Tools:SNMP MIB Browser` menu item.
- Make sure the application was built to monitor the device you have selected. The problem may be that the application was built to monitor a different device than the one you are trying to monitor. For example, the application was built for an HP-UX agent, and you are trying to run that application on a Fred Router from Flintstones Company.
- Verify that the management station can talk to the agent. Use the `Tools:SNMP MIB Browser` menu item to verify that the manager station can actually talk to the agent and that the agent is responding with the values. Also, use the `Fault:Test IP/TCP/SNMP` menu item.
- Verify that the community name is set up properly in the `Options:SNMP Configuration` menu item. If you get a message indicating no response from the device, the management station may not have the proper community name defined for that device.
- Make sure that the version of the MIB loaded into NNM matches the MIB version that is implemented on the device being managed. When you get a new device on your network or upgrade an existing one, make sure that you get the MIB for that device.
- Verify that the following three files created by the MIB Application Builder are intact:
  - The registration file in `install_dir\registration\C\ovmib ($OV_REGISTRATION/$LANG/ovmib`, where `$LANG` is the language NNM is set to use). Make sure the registration file name for the application is the same as the application name that was specified when the application was created.

- The help file name is the same as the application name that was specified when the application was created.

The help file is:

*Windows:* `install_dir\help\C\ovmib\OVW\Functions`

*UNIX:* `$OV_HELP/$LANG/ovmib/OVW/Functions`

- On UNIX operating systems only, the line entry in the help index file in the `$OV_HELP/$LANG/OVW/functions/INDEX` directory corresponds to your specific MIB application.
- If you typed the object identifier in the MIB selection field, make sure you enter a full object identifier. For example, you cannot create a form application for `.iso.org.dod.internet`. The object ID must correspond to an object supported by the selected device.
- If you configured your Application Builder program to query DMI devices using the corresponding mapped MIB values, and there is no response from the DMI devices, do one of the following:
  - Use the DMI Browser feature of NNM.
  - Install an SNMP agent along with a DMI-to-SNMP mapping agent on the DMI device.

## Configuring Events

If you have problems configuring events using the `Options:Event Configuration` menu item, first see the vendor documentation that describes the traps generated by that specific device to make sure the enterprise and trap number correspond to what you have indicated.

To get a description of the NNM events shipped with the product, use the `Options:Event Configuration` menu item. From the `Event Configuration` dialog box, select `OpenView` as the `Enterprise Name`, then select an event name and select `Edit:Events->Describe` (`Describe Event`).

If you have defined an action for an event, but the action is never taken or it is not executed the way it was intended to, check the following:

- Verify the configuration. When you save the configuration for a new event, the entire `trapd.conf` file is rewritten, and an event is sent to `xnmevents`, `ovactiond`, and `pmd`. Common configuration problems are:

- The command is typed wrong.
- The arguments are not specified properly.
- Make sure the format message for a particular trap is intact. For example, if the format message for the `coldStart` trap has been erased, you will not see that trap in the event log file or the Alarm Browser window.
- Make sure you do not edit the `trapd.conf` file. The `trapd.conf` file contains the logging and action definitions. If this file is corrupted, the events you have configured may not be logged.
- Use `ovdumpevents` to see if the trap was decoded properly. Each time NNM receives an event, the `pmd` service logs an entry in the event database. For example, the event database identifies ASN.1 coding problems in enterprise-specific traps.
- Verify that the event is logged to the expected category. For example, if you selected `log only`, you will not see the event in the event browser window. If you selected `ignore`, you will not see the event.
- On UNIX operating systems only, compare your `$OV_CONF/C/trapd.conf` file with the default configuration file, `/usr/OV/newconfig/OVEVENT-MIN/conf/C/trapd.conf`.

## Event Reduction Capabilities

Event reduction is the process by which you can identify relationships between events. Once identified, a smaller number of new events with the same or higher information content can be generated.

Two event reduction strategies that NNM uses are ECS correlations and Correlation Composer correlators. For more information about these strategies, see “ECS Correlations” on page 349 or “Correlation Composer Correlators” on page 371.

### Troubleshooting ECS Correlations

If you are having problems with any of the ECS correlations, see “Troubleshooting” on page 366 for more information.

### Troubleshooting Correlation Composer Correlators

If you are having problems with any of the Correlation Composer correlators, see “Troubleshooting” on page 381 for more information.

## Loading Internet MIBs

When you load an Internet MIB, the `Options:Load/Unload MIBs:SNMP` menu item takes all vendors' Internet MIBs and puts them into one single loaded MIB database, `install_dir\conf\($OV_CONF) snmpmib`. The `Load/Unload MIBs` menu item creates a binary file, `install_dir\conf\snmpmib.bin ($OV_CONF/snmpmib.bin)`. The SNMP MIB Browser, MIB Data Collection, MIB Application Builder, and the MIB applications built using the MIB Application Builder access the loaded MIB database for their routines.

---

### NOTE

Always use the `Options:Load/Unload MIBs:SNMP` menu item to modify the loaded MIB database; do *not* edit the loaded MIB database directly.

---

If you have problems including Internet MIBs in the MIB description file using the `Options:Load/Unload MIBs:SNMP` menu item, check the following:

- Make sure that the format of the MIB definition is correct. For example, it must have an ASN.1 module `DEFINITIONS` clause. The enterprise-specific files should only contain modules that are unique to that vendor.

The format must conform to *RFC 1155*, *RFC 1212*, *RFC 1902*, *RFC 1903* or *RFC 1904*. *RFC 1155* and *RFC 1212* apply to the SNMPv1 SMI, while *RFC 1902*, *RFC 1903*, and *RFC 1904* apply to the SNMPv2 SMI.

- Make sure that the syntax is correct. Hewlett-Packard has verified that the syntax of the MIB modules included with the NNM product is correct. However, if you get the MIB module directly from the vendor, you may have to correct syntax errors. The `Options:Load/Unload MIBs:SNMP` menu item checks for syntax errors and displays an error that tells you where the error is and how to fix it. The most common syntax errors are typing errors and non-conformance to the RFCs.
- Compare the MIB with the Hewlett-Packard MIB to look for errors. By comparing the MIB that does not load with the Hewlett-Packard MIB, you may be able to troubleshoot the problem. The Hewlett-Packard MIB loads correctly.

## Known Limitations

The object label in a MIB module must be unique. Due to lack of standards this is not clearly identified by SNMP. If you have mnemonic names for an object ID component that is used elsewhere in the module, the second object shows up in the MIB tree under the first object that has the common component name. Furthermore, the translation from the mnemonic object name to the numeric object ID will be incorrect, causing SNMP queries to fail.

For example, suppose the xyzCorp has the enterprise MIB device object defined in a module XYZCORP-MIB under  
.iso.org.dod.internet.private. as

```
enterprises.xyzCorp.bridge.device.name
                                   .number
                                   .ROMid
```

and another device object defined as

```
enterprises.xyzCorp.hub.device.model
                                   .serialNumber
                                   .otherInfo
```

The MIB parser cannot distinguish between the two device objects. As a result, the xyzCorp fields model, serialNumber, and otherInfo will appear incorrectly in the Internet MIB tree as

```
enterprises.xyzCorp.bridge.device.model
                                   .serialNumber
                                   .otherInfo
```

To circumvent the problem, patch the enterprise-specific MIB to use a unique component name. For this example, change .hub.device to .hub.hubdevice. When the object name is translated to the numeric object ID used in the SNMP query, the proper object ID will be used and the SNMP query should work.

A second (and more contrived) alternative is to put the second and any subsequent occurrences of duplicate object labels (and any objects based on them) in a separate ASN.1 module with a fully-qualified reference as to their proper position in the global Internet MIB tree.

For example, the first occurrence of device could be defined in a module as

```
XYZCORP1-MIB DEFINITIONS ::= BEGIN
xyzCorp OBJECT IDENTIFIER ::= { iso org(3) dod(6) internet(1) private(4)
                                   enterprises(1) 999 }
```

```

bridge OBJECT IDENTIFIER ::= { xyzCorp 1 }
device OBJECT IDENTIFIER ::= { bridge 1 }
-- all subsequent children...
.
.
.
END

```

and the second in a module as

```

XYZCORP2-MIB DEFINITIONS ::= BEGIN
hub OBJECT IDENTIFIER ::= { iso org(3) dod(6) internet(1) private(4)
                               enterprises(1) xyzCorp(999) }
device OBJECT IDENTIFIER ::= { hub 1 }
-- all subsequent children...
.
.
.
END

```

## X Windows Components (UNIX operating systems only)

This section discusses X Windows-related files for NNM only. If the problem does not appear to be with this product, refer to your X Windows documentation for further troubleshooting information.

To verify the status of the background services, run the `ovstatus` command. For more information about `ovstatus` and how to troubleshoot the background services, see “Troubleshooting Background Services” on page 544 in this appendix.

To determine whether there is a problem with the X Windows component of NNM, try restoring the default X resources for this product, and see if this solves the problem.

1. Ensure that the files in the `$APP_DEFS` directory contain the installed X resource defaults for the product: `XNm`, `OVW`, `XNmevents`, `XNmgraph`, `OVHelp`, etc.

The X resource defaults are listed in the `ovw` and `ovhelp` manpages. The `XNm` resource defaults are listed in the `XNm` resource file itself.

2. Ensure that the `XENVIRONMENT` shell variable is not set.

3. Remove resources starting with `XNm`, `XNmevents`, `XNmgraph`, `OVw`, or `ovhelp` from your `$HOME/.Xdefaults` file. Or, temporarily rename this file.
4. Restart X Windows.
5. Restart `ovw` (`$OV_BIN/ovw`) without any X resources specified as arguments.

## Online Help

The text files used for the NNM Help operations (not including reference pages or UNIX manpages) are below the directory `install_dir\help\C\help\C` (`$OV_HELP/help/C`). On UNIX operating systems only, you can also use the `man` command on the manpage name.

## Troubleshooting Windows Applications

This section discusses troubleshooting applications on Windows operating system.

### Problems Launching Applications

If you experience problems launching the Event Viewer, Performance Monitor, Registry Editor, or Windows Diagnostics, and an error message like `Permission denied` or `Access denied` appears after launching the application, it may be that you do not have enough privileges to be able to administer or view the remote computer's properties.

### Windows Operating System Tools

The Windows operating system tools, listed below, can assist you in troubleshooting Windows applications. These tools appear on NNM's menus, depending on the applications on your system. There can be up to eight Windows tools menu items:

- `Fault:Windows Event Viewer` — always present
- `Performance:Windows Performance Monitor` — always present
- `Configuration:Windows Registry Editor` — present only if the currently logged-on user has Administrative privileges
- `Fault:Windows Diagnostics`
- `Server Manager and User Manager` — present only if running on a Windows server (as opposed to a Windows workstation)
- `SMS Properties and Run SMS` — present only if SMS Administrator software has been installed

If you launch SMS and do not see an OpenView icon when viewing the properties on a node, verify that NNM is running.

If you have installed SMS Administrator software on your system and the two SMS-related menu items are missing, ensure that you installed the correct SMS version. See NNM's release notes for more information.



In order to be able to view SMS properties for a node selected on the NNM map, you must copy the `machview.exe` utility from your SMS CD-ROM to the `smsadmin\x86.bin` directory where SMS Administrator software is installed.

---

## Recommended Logging and Tracing Practices

The following sections discuss the logging and tracing practices of Network Node Manager.

### Logging and Tracing

Table B-3 lists the two logging and tracing facilities used by NNM.

**Table B-3**                    **Logging and Tracing Facilities**

For this service	Use this facility
pmd netmon ovactiond ovalarmsrv ovcapsd ovrepld ovrequestd snmpCollect	Command line options.  See the reference pages in NNM's online help (or the UNIX manpages) for the command options for these background services.
ipmap ovspmd ovtopmd ovuispmd ovwdb	Windows Event Viewer. Also see the <i>ovtracelog</i> reference page in NNM's online help (or the UNIX manpage).  nettl for logging and tracing.  netfmt to format and view the logged data.

---

**NOTE**

Tracing is normally used only by support personnel. Tracing, especially in the case of *netmon*, can cause the trace file to grow extremely large (multiple megabytes in size) fairly quickly. If you use tracing, remember to monitor the size of the trace file frequently, and turn tracing off as soon as you are finished.

---

## Using Command Line Options

The following section describes logging and tracing options for the NNM background services `pmd`, `netmon`, `ovactiond`, `snmpCollect`, and `ovalarmsrv`.

The `install_dir\bin\pmd` (`$OV_BIN/pmd`) event logging facility has the following characteristics:

- The default logging is set to ON. Specify the `n` option in the `pmd.lrf` to prevent event logging.
- The event log database is, by default, set to 16 MB. When this space is used up, the oldest log is truncated, and new events are written into the reclaimed space. Refer to the `ov_event` reference page in NNM's online help (or the UNIX manpage) for more information.

The `install_dir\bin $OV_BIN/pmd` background service logging facility has the following characteristics:

- The default logging is set to log WARNING, ERROR, and DISASTER.
- The log file is `install_dir\log\pmd.log0` (`$OV_LOG/pmd.log0`). When this log file is filled to a user-configurable size, its contents are automatically moved to the file `install_dir\log\pmd.log1` (`$OV_LOG/pmd.log1`).

The `install_dir\bin\pmd` (`$OV_BIN/pmd`) event and service tracing facility has the following characteristics:

- The default tracing is set to OFF. Specify the `T` option in the `pmd.lrf` file to enable tracing.
- The trace file is `install_dir\log\pmd.trc0` (`$OV_LOG/pmd.trc0`). When this trace file is filled to a user-configurable size, its contents are automatically moved to the file `install_dir\log\pmd.trc1` (`$OV_LOG/pmd.trc1`).
- When tracing is turned on, event traces are also found in the trace file.

The `install_dir\bin\netmon` (`$OV_BIN/netmon`) tracing facility has the following characteristics:

- The default tracing is set to OFF.

- The default trace and hexdump file is `install_dir\log\netmon.trace` (`$OV_LOG/netmon.trace`).
- To turn tracing on, run `netmon -M tracemask` at the command line, or add the `-m tracemask` option to `netmon.lrf`.
- To turn tracing off, specify `netmon -M 0` (zero).

The following information explains the variable `tracemask` discussed in the `netmon` tracing facility. Trace masks specify the type of output listed in `netmon`'s trace file. To select multiple output types, add the individual `tracemask` values together and enter the sum. `netmon` has the following tracing options:

0	Turn off tracing.
1	Trace ICMP echo requests.
2	Trace ICMP echo replies and timeouts.
4	Trace SNMP requests.
8	Trace SNMP replies and timeouts.
16	Trace traps generated.
32	Trace traps received.

Some examples follow. These assume `netmon` is already running.

To trace ICMP echo replies and timeouts, enter the following command:

```
netmon -M 2
```

To trace ICMP echo replies and timeouts, as well as SNMP replies and timeouts, add their `tracemask` values (2 + 8) to determine the `tracemask` value to enter:

```
netmon -M 10
```

The `install_dir\bin\ovactiond` (`$OV_BIN/ovactiond`) tracing facility has the following characteristics:

- The default tracing is set to ON.
- The default tracing file is `install_dir\log\ovactiond.log` (`$OV_LOG/ovactiond.log`) and `install_dir\log\ovactiond.log.old` (`$OV_LOG/ovactiond.log.old`).
- To turn verbose tracing on and off when tracing is on, specify `ovactiond -V` (toggle ON/OFF).

- The `-t` option must have a value greater than 0 (zero) to see the return value of the event actions.

The `install_dir\bin\ovalarmsrv` (`$OV_BIN/ovalarmsrv`) tracing facility has the following characteristics:

- By default, logs only errors.
- Logs errors to `install_dir\log` (`$OV_LOG`) `ovalarmsrv.trace`.

To change the level of logging that is performed, edit the LRF file and add `-D2` as a parameter. This will increase the logging information (for example, add tracing information as well).

The `install_dir\bin\snmpCollect` (`$OV_BIN/snmpCollect`) logging facility has the following characteristics:

- The default logging is set to ON.
- The data that is collected is stored in the file `install_dir\log\snmpCol.trace` (`$OV_LOG/snmpCol.trace`).

The `install_dir\bin\snmpCollect` (`$OV_BIN/snmpCollect`) tracing facility has the following characteristics:

- The default tracing is set to OFF.
- The default tracing is limited.
- The default trace file is `install_dir\log` (`$OV_LOG`) `snmpCol.trace` and `install_dir\log` (`$OV_LOG`) `snmpCol.trace.old`.
- To turn the tracing on and off, specify `snmpCollect -T` (toggle ON/OFF) while `snmpCollect` is running.
- To turn verbose tracing on and off when tracing is on, specify `snmpCollect -V` (toggle ON/OFF) while `snmpCollect` is running.
- To trace `snmpCollect` initialization, add the options `-T -V` to `install_dir\lrf\snmpCollect.lrf` (`$OV_LRF/snmpCollect.lrf`). This turns off verbose tracing from the moment `snmpCollect` starts up.

## Using the Windows Application Event Log

NNM services perform logging (to capture network activities such as state changes, errors and connections) and tracing (to capture inbound and outbound packets going through the network, as well as loopback or header information). All tracing and logging is directed to the Windows

Event Log and can be accessed through the Windows Event Viewer application. See the *ovtracelog* reference page in NNM's online help (or the UNIX manpage) for further details. You can use the following troubleshooting tools:

- **tracing**, which records a detailed record of internal events taking place within the software. Tracing of the NNM background services is generally only used on the advice of HP Support to diagnose difficult problems, with the aid of the source code. API tracing is used by developers to determine whether they are using the HP OpenView APIs correctly.
- **logging**, which records externally-observable events such as connection establishment or termination of child services, and errors, which background services ordinarily have no other way of reporting.

Users should run tracing only when something is seriously wrong, and only on the advice of your HP support representative when it is necessary to reconstruct in detail the sequence of events that caused the problem. The trace output is intended to be interpreted by the developers of the NNM product, with the source code at hand. It is not intended to be meaningful to the ordinary user.

### Using `nettl` and `netfmt` (UNIX operating systems only)

`nettl` and `netfmt` enable system administrators and developers to gather and view data about the operation of NNM software. `nettl` is the administrative interface to the tracing and logging facilities discussed previously.

It is recommended that you run the `nettl` logging facility at all times. This way, you can view the constantly updated log file to help you determine what happened when something goes wrong. Also, paradoxically, performance is better when logging is turned *on*.

`netfmt` formats the log and trace output into human-readable form.

Services use `nettl` to log or trace in particular subsystems. To examine the log or trace output of a service, you need to know the name of the subsystem the service uses. Table B-4 shows the subsystems and services.

---

**NOTE**

The `nettl` and `netfmt` commands are in `$OV_BIN`. For example, to run the `nettl -status` command, you would type:

`$OV_BIN/nettl -status`

**Table B-4 Subsystem Services**

Subsystem	Services Using this Subsystem	Purpose of Service
OVS	ovsppmd	Starts and stops NNM background services.
OVW	ovwdb, ipmap	Provides access to object database. Automatically draws IP topology maps.
OVEXTERNAL	ovuisppmd; ovtppmd; also any third-party-developed software using the OVuTL facility.	Maintains the topology database.
OVWAPI	Any software using the OVw (HP OpenView Windows) API.	Provides programmatic application integration points.

**Capturing Logging and Tracing Output with nettl (UNIX operating systems only)**

1. To determine whether the logging/tracing facilities are currently running, type the following command as `root`:

**nettl -status**

The output includes the log and trace status for each active subsystem.

2. If necessary, use the `nettl` command to start the logging/tracing facilities. As `root`, type the following command:

**nettl -start**

This command initializes the logging/tracing facilities and starts the default logging. By default, the `OVS`, `OVW`, `OVWAPI`, and `OVEXTERNAL` subsystems log error and disaster messages; the `OVS` subsystem also logs informative messages.

3. Once `nettl` is running, you can use command options to customize the kind of data you want to capture for any of the `nettl` subsystems shown in the previous table. The next section contains examples of various `nettl` commands.

**Examples of nettl Commands** Use the following examples to get started. After you have captured the data you want, go to “Formatting Network Data with netfmt” on page 612 in this section.

- **Logging Facility** - Turning it on.

There are four classes of logging messages: informative (i), warning (w), error (e), and disaster (d). Default logging always includes disaster-class messages; you can turn on any of the other three classes by typing a `nettl` command for a specific subsystem.

The command syntax is:

```
nettl -log class -entity subsystem OVEXTERNAL
-log class
```

Specifies the classes of logging to be turned on for the specified subsystems.

```
-entity subsystem
```

Specifies the subsystems for which logging is to be modified. *For logging done by HP OpenView products, OVEXTERNAL must always be in the list.*

**Example 1:**

Turn on logging of informative messages for the `ovw` subsystem. As `root`, type the following command:

```
nettl -log i -entity OVW OVEXTERNAL
```

Note that this command will turn off any log classes not listed in the command (in this example, the warning and error messages would be turned off). The one exception is disaster logging, which cannot be turned off.

**Example 2:**



Turn on logging of informative, warning, and error messages for the OVS subsystem. As root, type the following command:

```
nettl -log i w e -entity OVS OVEXTERNAL
```

- **Tracing Facility** - Turning it on.

There are eight kinds of tracing that nettl can turn on, as shown in the following table.

**Table B-5 Tracing Facilities Activated by nettl**

Type of Tracing	Keyword	Mask Number
inbound header tracing	hdrin	0x80000000
outbound header tracing	hdrout	0x40000000
inbound PDU tracing	pduin	0x20000000
outbound PDU tracing	pduout	0x10000000
procedure entry/exit tracing	proc	0x08000000
state machine tracing	state	0x04000000
error tracing	error	0x02000000
log call tracing	logging	0x01000000

The command syntax is:

```
nettl -traceon kind -entity subsystem OVEXTERNAL -file  
filename
```

```
-traceon kind
```

Specifies the kinds of tracing to be turned on for the specified subsystems.

```
-entity subsystem
```

Specifies the subsystems for which tracing is to be turned on. *For tracing done by HP OpenView products, OVEXTERNAL must always be in the list.*

```
-file filename
```

Specifies the prefix of the trace file you want to create. Trace output will be written to *file*.TRC0 and *file*.TRC1. (See “How nettl Generates the Binary Data Files” on page 611 for a description of these two trace files.)

**Example 1:**

Do error tracing for the OVW subsystem, putting the data in the trace files, /tmp/nettl.TR0 and /tmp/nettl.TR1. As root, type the following all on one line:

```
nettl -traceon error -entity OVW OVEXTERNAL -file /tmp/nettl
```

**Example 2:**

Do both inbound and outbound PDU tracing for the ovw subsystem, putting the data in the trace files, /tmp/nettl.TR0 and /tmp/nettl.TR1. To enable more than one kind of tracing, either specify all the appropriate keywords, all the appropriate mask numbers, or the logical OR of the mask numbers. As root, type one of the following command variations:

```
nettl -traceon pduin pduout -entity OVW OVEXTERNAL -file /tmp/nettl
```

or

```
nettl -traceon 0x20000000 0x10000000 -entity OVW OVEXTERNAL -file /tmp/nettl
```

or

```
nettl -traceon 0x30000000 -entity OVW OVEXTERNAL -file /tmp/nettl
```

---

**NOTE**

Do not leave off the `-file` argument to `nettl -trace on`! If you do, unformatted trace output will be sent to your screen.

- **Tracing Facility** - Turning it off.

While logging should always remain on, you should run the tracing facility only when a serious problem occurs, and it is necessary to reconstruct in detail the sequence of events that caused the problem. After you have completed a trace, you should turn tracing off.

The command syntax is:

```
nettl -traceoff -entity subsystem
```

`-traceoff` Specifies that tracing is to be turned off for the specified subsystems.

```
-entity subsystem
```

Specifies the subsystems for which tracing is to be turned off. To turn tracing off completely, specify all subsystems for which tracing is currently on.

Note that you cannot turn off trace kinds individually. To change the trace kinds enabled for a subsystem, turn tracing off for that subsystem, then turn tracing back on for the particular kinds in which you are interested.

### Example 1:

Disable tracing for the `OVEXTERNAL` subsystem. As `root`, type the following command:

```
nettl -traceoff -entity OVEXTERNAL
```

- **Logging and Tracing Facilities - Turning both off.**

To stop the logging/tracing facilities, as `root`, type the following command:

```
nettl -stop
```

**How nettl Generates the Binary Data Files** The first time you run a `nettl` command for capturing logging data, the log file `nettl.LOG00` is generated. Binary data captured from each successive `nettl` logging command is appended to this file, until it is filled. Then a second log file with the suffix `.LOG01` is created in which new data is stored. When this second file fills up, the `.LOG00` file is overwritten.

When you run a `nettl` command for capturing trace data, you specify a filename prefix, to which the suffix `.TRC0` is added. For each successive `nettl` tracing command, you can have the binary data appended to this file, or you can specify a new file in which to store the data. If you append binary data from successive traces to the same file, it will eventually fill up. A new file with the same prefix and the suffix `.TRC1` will then be created for storing new data. If this second file fills up, then a new trace file with the same prefix and the `.TRC0` suffix is created.

If you need to check the name of a log file before formatting it with `netfmt`, use a `nettl -status` command.

**Formatting Network Data with netfmt** The command syntax is:

```
netfmt -f binary_file  
-f binary_file
```

Specifies the log or trace file containing the binary data

It is possible to do elaborate filtering of the output, as `netfmt` has many other options that cannot be described here. Some of these options, for example, let you specify the output of a log or trace file to be filtered by timestamp, subsystem, or message class. UNIX only: for more information, see the `netfmt` manpage.

**Examples of netfmt Commands** The following example will help you start using the `netfmt` command.

To format the log file, `$NETFMT_LOG_FILE`, type

```
$NETFMT/netfmt -f $NETFMT_LOG_FILE | more
```

The next section shows what the output of `netfmt` looks like.

**Log and Trace File Output** Below are two complete examples of using the `nettl` and `netfmt` commands, and the final output you might see.

**Example 1:**

1. You ran a `nettl` logging command for informative messages in the OVW subsystem, in order to observe the startup of the `ovspmd` service, by typing:

```
nettl -log i e d -entity OVS OVEXTERNAL
```

The binary data was stored in the `$NETFMT_LOG_FILE` log file.

2. You formatted and displayed the binary log file by typing:

```
netfmt -f $NETFMT_LOG_FILE | more
```

The output would be similar to the following:

```
*****OpenView*****@#%  
Timestamp      : Wed Nov 17 1995 17:28:58.441517  
Process ID     : 22756           Subsystem      : OVS  
User ID ( UID ) : 0             Log Class      : INFORMATIVE  
Device ID      : -1            Path ID        : -1  
Connection ID  : -1            Log Instance   : 0
```

```
Software          : ovspmd
Hostname          : pinetum.cnd.hp.com
~~~~~

Initialized successfully:
ovspmd

*****OpenView*****@#%

Timestamp        : Wed Nov 17 1995 17:28:59.352086
Process ID       : 22756           Subsystem        : OVS
User ID ( UID )  : 0             Log Class        : INFORMATIVE
Device ID        : -1            Path ID          : -1
Connection ID    : -1            Log Instance     : 0
Software         : ovspmd
Hostname        : pinetum.cnd.hp.com
~~~~~

Accepted request connection (fd 5)

*****OpenView*****@#%

Timestamp        : Wed Nov 17 1995 17:28:59.355761
Process ID       : 22756           Subsystem        : OVS
User ID ( UID )  : 0             Log Class        : INFORMATIVE
Device ID        : -1            Path ID          : -1
Connection ID    : -1            Log Instance     : 0
Software         : ovspmd
Hostname        : pinetum.cnd.hp.com
~~~~~

Received request, fd 5
[9] [START] ""

*****OpenView*****@#%

Timestamp        : Wed Nov 17 1995 17:28:59.358398
Process ID       : 22756           Subsystem        : OVS
User ID ( UID )  : 0             Log Class        : INFORMATIVE
Device ID        : -1            Path ID          : -1
Connection ID    : -1            Log Instance     : 0
Software         : ovspmd
Hostname        : pinetum.cnd.hp.com
~~~~~

Building master M table from $OV_CONF/ovsuf

*****OpenView*****@#%

Timestamp        : Wed Nov 17 1995 17:28:59.366810
Process ID       : 22756           Subsystem        : OVS
User ID ( UID )  : 0             Log Class        : INFORMATIVE
Device ID        : -1            Path ID          : -1
Connection ID    : -1            Log Instance     : 0
```

## Troubleshooting NNM Itself

### Recommended Logging and Tracing Practices

```
Software      : ovspmd
Hostname      : pinetum.cnd.hp.com
~~~~~
Reporting M status, code = SUCCESS
  object manager name: OVSPMD
  behavior:          OVSPMD
  state:             RUNNING
  PID:               22756
  exit status:       -
```

#### Example 2:

1. You ran a `nettl` tracing command for procedure tracing in the `OVEXTERNAL` subsystem, in order to observe some procedure calls in `ipmap`, by typing:

```
nettl -traceon proc -entity OVEXTERNAL -file $NETFMT_LOG_FILE
```

The binary data was stored in the trace file `$NETTL_TRC_FILE`.

2. You formatted the binary trace file by typing:

```
netfmt -f $NETTL_TRC_FILE > outfile
```

A brief fragment of the output appears below:

```
*****OpenView*****@#%
Timestamp      : Wed Nov 17 1993  18048:44.443027
Process ID     : 23059             Subsystem      : OVW
User ID ( UID ) : 214             Trace Kind     : Proc. Entry/Exit
Device ID      : -1               Path ID        : -1
Connection ID  : -1
Software       : $OV_BIN/ipmap
Hostname       : pinetum.cnd.hp.com
~~~~~
FieldBinding::FieldBinding(OVwFieldBinding) entered
*****OpenView*****@#%
Timestamp      : Wed Nov 17 1993  18:04:44.444646
Process ID     : 23059             Subsystem      : OVW
User ID ( UID ) : 214             Trace Kind     : Proc. Entry/Exit
Device ID      : -1               Path ID        : -1
Connection ID  : -1
Software       : $OV_BIN/ipmap
Hostname       : pinetum.cnd.hp.com
~~~~~
FieldBinding::init() entered
*****OpenView*****@#%
```

```
Timestamp      : Wed Nov 17 1993  18:04:44.446053
Process ID     : 23059             Subsystem      : OVW
User ID ( UID ) : 214             Trace Kind     : Proc. Entry/Exit
Device ID      : -1              Path ID        : -1
Connection ID  : -1
Software       : $OV_BIN/ipmap
Hostname       : pinetum.cnd.hp.com
~~~~~
FieldBinding::init() exit
*****OpenView*****@#%
Timestamp      : Wed Nov 17 1993  18:04:44.447404
Process ID     : 23059             Subsystem      : OVW
User ID ( UID ) : 214             Trace Kind     : Proc. Entry/Exit
Device ID      : -1              Path ID        : -1
Connection ID  : -1
Software       : $OV_BIN/ipmap
Hostname       : pinetum.cnd.hp.com
~~~~~
FieldBinding::copyFieldValue() entered
*****OpenView*****@#%
Timestamp      : Wed Nov 17 1993  18:04:44.448754
Process ID     : 23059             Subsystem      : OVW
User ID ( UID ) : 214             Trace Kind     : Proc. Entry/Exit
Device ID      : -1              Path ID        : -1
Connection ID  : -1
Software       : $OV_BIN/ipmap
Hostname       : pinetum.cnd.hp.com
~~~~~
FieldBinding::copyFieldValue() exit
*****OpenView*****@#%
Timestamp      : Wed Nov 17 1993  18:04:44.450233
Process ID     : 23059             Subsystem      : OVW
User ID ( UID ) : 214             Trace Kind     : Proc. Entry/Exit
Device ID      : -1              Path ID        : -1
Connection ID  : -1
Software       : $OV_BIN/ipmap
Hostname       : pinetum.cnd.hp.com
~~~~~
FieldBinding::FieldBinding(OVwFieldBinding) exit
```

## Web Launcher Error Log

The Launcher error log records internal errors from the Launcher. You can use the output from this file when discussing a problem with HP support personnel.

The Launcher error log file is located in:

*Windows:* `install_dir\www\logs\launcher\error_log`

*UNIX:* `/var/opt/OV/www/launcher/error_log`

The error log contains the following information:

- Date
- Host
- User
- Session
- Reason or error

The error log file continues to grow without bounds. You need to watch the size of this file and truncate it as needed.

If the error log file does not give you sufficient information for troubleshooting, you can set a parameter in the URL for `ovlaunch.exe` that will enable more in-depth error logging on a session level.

The parameter is:

`http://hostname:[port]/OvCgi/ovlaunch.exe?Debug=/tmp/file`

The output of this file is less “friendly” than other error log files, but it contains valuable information that may be useful to an HP support person who is trying to assess the problem.



## Improving Traffic Management and Performance

### Traffic Management

Status polling generates the most network traffic. You can monitor the traffic generated by NNM by using the `netstat -s` command. Run the `netstat -s` command several times and observe the differences in the number of packets sent and received. Be sure that the management station is not generating any other outbound traffic. Note that the command used to show the number of packets sent and received may be different depending on the operating system you use. Make sure that you are using the appropriate command.

You can regulate the traffic generated by NNM through the following methods:

- Adjust polling intervals through the `Options:SNMP Configuration` or `Options:Network Polling Configuration` menu item. Reducing the amount of polling traffic by lengthening polling intervals may delay real-time map updates, resulting in a less accurate map.
- Apply a discovery filter to ensure that you are monitoring only what needs to be monitored.
- Unmanage nodes with the `Edit:Unmanage Objects` menu item such that you are monitoring only what really needs to be monitored. Note that this tactic decreases the amount of available management information for unmanaged nodes.
- Adjust the amount of MIB data that is collected through the `Options:Data Collection & Thresholds:SNMP` menu item. You can:
  - 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 either deleting nodes, deleting instances, excluding nodes, or changing polling intervals.

If NNM starts to use excessive swap space, performance may decrease. To check the size of `ovw` (and other services), use `Help:About HP OpenView`, and click `More Info`. If the service grows too large, exit NNM and restart it. (The program automatically saves the map between invocations.)

`netmon` will generate some network broadcast traffic; how much traffic depends on several factors.

- If you use `netmon` without the `-J` option, `netmon` generates, if necessary, just an ARP Request. There are several factors that determine whether `netmon` needs to generate an ARP Request.
  - For nodes that are up, but not on the same subnet where the management station is, `netmon` generates just one ICMP echo request packet and gets back one response. In this case, no ARP Request is generated.

Because the management station is on a different subnet, a request will go through a gateway. Gateways always have large ARP caches. `netmon` knows the gateway's link-level address and does not have to broadcast an ARP Request to that gateway. `netmon` only has to send one ICMP echo request through the gateway and gets back one response.
  - For devices that are on the same subnet as the management station, `netmon` may generate just one ICMP echo request, or it may also generate one ARP Request as well. Because the nodes are on the same subnet, `netmon` does not have to go through a gateway. Broadcasting an ARP Request will depend on whether or not `netmon` has in its own ARP cache in the device's link-level address.
  - `netmon` will generate an ARP Request based on whether the address that is being polled for status is still in the ARP cache of that system.

The presence of the address in the ARP cache is affected by how many nodes are on the subnet, and by how much physical memory is in the system (some systems will automatically adjust the size of their ARP table).
- If invoked with the `-J` option, `netmon` may cause up to three broadcast ICMP echo requests to be broadcast on each subnet it discovers. These broadcasts will only be issued on a subnet if:

- No broadcast ICMP echo request has ever been issued on this subnet on behalf of this management station.
- An agent capable of issuing a broadcast ICMP echo request on netmon's behalf has been discovered on this subnet. All HP-UX SNMP agents have this capability.
- The amount of netmon's broadcast traffic also depends on your network's configuration, including:
  - How many nodes are up and how many are down.
  - How many nodes are on the same subnet.
  - How many nodes are on other subnets.
  - How large the permanent ARP cache is.
  - How probable it is that a particular IP address can be found in the permanent ARP cache.
- For IPX discovery, netmon uses broadcast IPX packets as the basis of its discovery. The following broadcasts occur for each discovery interval:
  - A single broadcast RIP request.
  - A single broadcast SAP request.
  - Two broadcast IPX diagnostic requests for each known IPX network, 10 seconds apart.
  - You can change or schedule the IPX discovery interval using `Options:Network Polling Configuration operation`

## Performance

The performance of NNM is limited primarily by three factors:

- The amount of memory installed in the manager system.
- The management station's processor (CPU) speed.
- The performance of the graphical display adapter (X Windows).

Minimum system requirements and memory recommendations are detailed in the *Performance and Configuration* guides for NNM.

One good way to find out if your management station has sufficient memory is to select the icon for your management station on the map. Then select the Performance:Network Polling Statistics menu item. If the graph shows negative numbers, the polling queue has fallen behind because of resource shortage.

Following are some tips to conserve memory and increase performance:

- You may want to install additional memory and/or upgrade your CPU.
- Reduce the number of devices that need to be monitored, thus decreasing the polling load. Refer to “Controlling the Amount of Traffic Generated by NNM” on page 163 for more information.
- Change the polling interval setting to a less frequent interval. Refer to “Troubleshooting Background Services” on page 544 for more information.
- If you are currently running OVW sessions directly on a management station, you can improve system performance by running OVW sessions on an NNM remote console instead.
- You can reduce NNM’s memory usage by changing `ipmap`’s persistence level to a higher level than `All Levels`. See *A Guide to Scalability and Distribution* for more information.
- Run `ipmap` with the `-D 2` option. This option prevents the automatic creation of node submaps for end nodes (unless an end node is a gateway). Nodes with submaps that have not been created are marked with an empty string on the label. See the `ipmap` reference page in NNM’s online help (or the UNIX manpage) for information on how to use the `- l` or `- L` options to modify this empty string. Double-clicking on such a node will result in the creation of the submap. The advantage of this option is that no memory is used for end node submaps that are never viewed. The disadvantage is that the locate operation will not work for addresses for end nodes.

The option can be set in:

*Windows:* `install_dir\registration\C\ipmap`

*UNIX:* `$OV_REGISTRATION/C/ipmap`

- Partition the environment into multiple maps. See “Partitioned Internet Submaps” on page 234 for more information on partitioning submaps.

- Delete unwanted nodes. After the nodes on a particular network have been discovered, you may want to delete some of the nodes, for example, PCs that are not being managed directly.
- Restart the system after active discovery. Some extra memory is consumed in the source of active discovery. After active discovery is finished, it may help to stop and restart the system to recover memory used during the discovery process.
- Use the on-demand submap feature, which may greatly reduce the amount of memory required. The drawback is that when opening transient submaps, some extra processor work will be expended and some delay may be noticeable.
- NNM on a Windows operating system performs best if you use DNS. If you are going to use DNS, enable DNS for Windows Resolution and configure DNS correctly. In particular, when installing NNM, the IP hostname and Windows computer name of the intended NNM Manager must be correct and identical.

### Performance on UNIX Operating Systems

On UNIX operating systems only, you can also increase performance in some cases by minimizing file system swapping. NNM performance is best when there is no file system swapping required. With no file swapping, `ovwdb` is running as efficiently as possible; all the object information it is managing is residing in physical memory. Any system swapping that must be done is affected by two factors: the number of objects maintained in physical memory, and the service size limit.

- If you have a system where NNM performance is poor due to file system swapping, you may use the `-n` option for `ovwdb` to limit the maximum number of objects that `ovwdb` can keep in physical memory, while storing the rest in a database.

Note that this option is not always an appropriate solution for memory problems due to file swapping, and it is not a good performance enhancement because `ovwdb` database queries are often random. For example, a `locate` operation will cause a query over many objects in searching for a field value. In this operation, using the `-n` option would cause `ovwdb` to update the cache with each object involved in the search, as well as with all the field values for those objects. The best way to correct poor performance due to file system

swapping is to increase the amount of physical memory available to `ovwdb`. More physical memory reduces the amount of swapping needed.

If the `-n` option is not used (the default state), `ovwdb` will continue to grow as new objects are created until either:

- The maximum service size limit is reached.
- The system swap space is exhausted.
- Another factor that can limit the number of objects managed by `ovwdb` is the service size limit. If the maximum service size limit is reached, `ovwdb` will terminate. For information on increasing the service size limit, see the appropriate UNIX operating system administration guide.

---

# **C**      **Changing All the Symbols for a Particular Device**

## Procedure

Each of the following steps is explained in more detail in the example beginning on page 625.

---

### NOTE

Back up any files before you modify them.

1. Create a *symbol registration file* to add a new symbol subclass.
2. Copy and modify a set of GIF graphic files for your new symbol and place the new set of GIF files into two locations.
3. Map a specific `sysObjectID` to the new symbol by editing the appropriate file within the `oid_to_sym_reg` directory structure.
4. Create a *field registration file* to define the appropriate capabilities for each new database field specified in the symbol registration file (step 1).
5. Force NNM to add the new field definitions to the database.
6. Ensure that the `sysObjectID` is listed in the `oid_to_type` file (or `HPoid2type` file) to provide information to NNM about the vendor and SNMP agent for the device you are mapping to the new symbol. This file can also be used to provide NNM with additional useful information about the device.
7. Force NNM to acknowledge your changes and update the map.
8. Open the NNM map and verify the changes.

Complete information about registration files is available in NNM's online manual, *Creating and Using Registration Files*.



---

## Example

In this example, you will make NNM display customized symbols for HP 9000 Series 700 workstations, rather than the predefined HP-UX workstation symbol.

If you follow this example, be sure that you back up any files before modifying them. You will modify the following files in this example:

- *Windows:*

```
install_dir\symbols\C\Computer\Computer_workstation
install_dir\www\htdocs\bitmaps\C\computer\workst.16.gif
install_dir\www\htdocs\bitmaps\C\computer\workst.20.gif
install_dir\www\htdocs\bitmaps\C\computer\workst.26.gif
install_dir\www\htdocs\bitmaps\C\computer\workst.32.gif
install_dir\www\htdocs\bitmaps\C\computer\workst.38.gif
install_dir\www\htdocs\bitmaps\C\computer\workst.44.gif
install_dir\www\htdocs\bitmaps\C\computer\workst.50.gif
install_dir\conf\oid_to_sym_reg\001_HP\002_HP_Other
install_dir\conf\oid_to_type
```

- *UNIX:*

```
$OV_www/htdocs/bitmaps/$LANG/computer/workst.16.gif
$OV_www/htdocs/bitmaps/$LANG/computer/workst.20.gif
$OV_www/htdocs/bitmaps/$LANG/computer/workst.26.gif
$OV_www/htdocs/bitmaps/$LANG/computer/workst.32.gif
$OV_www/htdocs/bitmaps/$LANG/computer/workst.38.gif
$OV_www/htdocs/bitmaps/$LANG/computer/workst.44.gif
$OV_www/htdocs/bitmaps/$LANG/computer/workst.50.gif
$OV_CONF/oid_to_sym_reg/001_HP/002_HP_Other
$OV_CONF/oid_to_type
```

## Create a Symbol Registration File

Because NNM provides a rich set of classes (shape outlines), you will rarely create an entirely new class of symbols. You almost always will be adding subclasses (graphics imposed over the shape) to an existing class.

To create a new subclass, create a new symbol registration file:

1. Copy the following file and name it **Computer\_workstation\_700**. Place the new file in the same directory as the original file:

- *Windows:*

```
install_dir\symbols\C\Computer\Computer_workstation
```

- *UNIX:*

```
$OV_SYMBOLS/$LANG/Computer/Computer_workstation
```

2. Edit the new symbol registration file named `Computer_workstation_700`. Modify the file to look like this:

```
SymbolType "Computer" : "HP700"
{
  Filebase "HP700";
  CursorSize 38;

  Capabilities {
    isHP700 = 1;
  }
}
```

Any file found in this directory is assumed to be a registration file. If you used a text editing program that makes an automatic backup of any file you edit, be sure to remove the backup file.

3. Change the permissions on the file to read-only.

## Copy and Modify the Symbol Graphics

You need a *set* of GIF files to provide multiple sizes of the same graphic so that when you resize the map, the graphics grow and shrink. The easiest way to create custom graphics is to copy an existing set and modify each file. Open NNM and select `Help:Display Legend` to see the existing collection of symbol graphics.

Each set of GIF files contains several files by the same name with different numbers. The numbers indicate the pixel size of the square GIF image (20x20, 50x50). For this example, use:

- *Windows:*

```
install_dir\www\htdocs\bitmaps\C\computer\workst.16.gif
```

```
install_dir\www\htdocs\bitmaps\C\computer\workst.20.gif
```

```
install_dir\www\htdocs\bitmaps\C\computer\workst.26.gif
```

```
install_dir\www\htdocs\bitmaps\C\computer\workst.32.gif
```

```
install_dir\www\htdocs\bitmaps\C\computer\workst.38.gif
```

```
install_dir\www\htdocs\bitmaps\C\computer\workst.44.gif
```

```
install_dir\www\htdocs\bitmaps\C\computer\workst.50.gif
```

- *UNIX:*

```
$OV_WWW/htdocs/bitmaps/$LANG/computer/workst.16.gif
```

```
$OV_WWW/htdocs/bitmaps/$LANG/computer/workst.20.gif
```

```
$OV_WWW/htdocs/bitmaps/$LANG/computer/workst.26.gif
```

```
$OV_WWW/htdocs/bitmaps/$LANG/computer/workst.32.gif
```

```
$OV_WWW/htdocs/bitmaps/$LANG/computer/workst.38.gif
```

```
$OV_WWW/htdocs/bitmaps/$LANG/computer/workst.44.gif
```

```
$OV_WWW/htdocs/bitmaps/$LANG/computer/workst.50.gif
```

1. Copy this set of `workst.##.gif` files and give them new meaningful names (for this example, use the name `HP700.##.gif`). Place them in the same directory as the ones you copied. Be sure to follow the naming scheme using the numbers in the file extensions
2. Change the permissions on the new GIF files so that you can modify them using the graphics program of your choice.  
  
Start with size 16. Because of size, this one is the hardest to modify. Write 700 inside the workstation's display screen in each GIF image.
3. Set the file permissions on the finished HP700 GIF files to read-only.
4. The graphic files for the symbols, themselves, are stored in two places: one directory used by NNM on the management station, and one directory used by the NNM web-based interface. (The web-based interface requires GIF files. Other additional graphic file formats are supported on the management station.)

**Example**

Place a copy of the new GIF files into the following directory for use by NNM on the management station. Don't worry if they are the only GIF files in this directory, because NNM ships with pixmaps:

*Windows:* `install_dir\bitmaps\C\computer`

*UNIX:* `$OV_BITMAPS/$LANG/computer`

---

**TIP**

Symbols in the web-based interface have the same appearance as in NNM on a management station, except that the symbol labels do not scale in the web-based interface. Instead, a long label is truncated to fit the space available.

---

## Map a sysObjectID to the New Symbol

NNM determines which symbol to draw on the map for a particular `sysObjectID` by referencing the appropriate file and file entries in the `oid_to_sym_reg` directory structure. For additional information, refer to the comments at the top of the appropriate file within the `oid_to_sym_reg` directory structure or the `oid_to_sym` reference page in NNM's online help (or the UNIX manpage).

---

**NOTE**

While you are editing one of these files, note the many-to-one relationship between `sysObjectIDs` and symbols. You can specify "leaf" values of `sysObjectIDs` in these files. For example, you can specify two `sysObjectIDs` separately:

```
1.3.6.1.4.1.11.2.3.2.2:Computer:HP-UX # 300
```

```
1.3.6.1.4.1.11.2.3.2.5:Computer:HP-UX # 700
```

A limited use of wildcards is allowed. Only a single `*` can exist as the last entry in a `sysObjectID` number sequence.

```
1.3.6.1.4.1.9.1.*:Computer:XX #Cisco devices use XX symbol
```

---

1. Locate the following file:

- *Windows:*  
`install_dir\conf\oid_to_sym_reg\001_HP\002_HP_Other`
- *UNIX:* `$OV_CONF/oid_to_sym_reg/001_HP/002_HP_Other`

2. Make a backup copy of the `002_HP_Other` file, if you have not already done so.

3. Edit the `002_HP_Other` file to modify the line referring to HP 9000 Series 700 workstations. Your changes map this device to the new HP700 subclass instead of the existing HP-UX workstation subclass.

Old line:

```
1.3.6.1.4.1.11.2.3.2.5:Computer:HP-UX #700
```

New line:

```
1.3.6.1.4.1.11.2.3.2.5:Computer:HP700 #700
```

## Define the Capabilities for Computer\_workstation\_700

Now create a *field registration file* to define the `isHP700` capabilities field referred to in your `Computer_workstation_700 symbol registration file`. Whenever the discovery and layout process creates a symbol of type `Computer:HP700`, it will automatically set the `isHP700` field to `TRUE`.

Complete information about field registration files is available in the NNM online manual, *Creating and Using Registration Files*.

Fields are defined in field registration files:

- *Windows:* `install_dir\fields\C\*.*`
- *UNIX:* `$OV_FIELDS/$LANG/*`

1. In this directory, create a new file called `HP700` and enter this information:

```
Field "isHP700" {
    Type Boolean;
    Flags Capability;
}
```

**Example**

Any file found in this directory is assumed to be a registration file. If you used a text editing program that makes an automatic backup of any file you edit, be sure to remove the backup file.

2. *UNIX operating systems only:* Make sure the file is readable. At the command prompt, type:

```
chmod 644 HP700
```

## Inform NNM about the New Fields

---

**NOTE**

Once you complete this section, the `isHP700` field will be added to the NNM database. It is nearly impossible to modify or remove a field once it is added, so proceed with care.

---

1. Make sure that your `PATH` environment variable includes:

- *Windows:* `install_dir\bin`
- *UNIX:* `$OV_BIN`

2. Log in as:

- *Windows:* Administrator
- *UNIX:* `root`

3. Close all NNM sessions (see “Closing All Current Sessions” on page 305).

4. Inform NNM of the new `isHP700` capability field by entering:

- *Windows:* **`ovw -fields`**

The following message indicates a successful update:

```
Created Boolean field "isHP700"
```

- *UNIX:* **`ovw -fields 2>&1 | tee /tmp/ovwFieldsOut`**

The following message in the `/tmp/ovwFieldsOut` file indicates a successful update:

```
/etc/opt/OV/share/fields/C/HP700: Created Boolean field "isHP700"
```

5. To check for syntax errors in both the *symbol registration file* and *field registration file* that you have created so far, at the command prompt, type:

```
ovw -verify
```

When this command is executed, NNM searches the predefined directories for registration files.

---

**NOTE**

---

You can specify directories other than the default directories by using environment variables, such as `OVwRegDir`, `OVwFieldDir`, and `OVwSymbolDir`.

For each registration file found, NNM executes the following steps:

- a. Opens the file
- b. Parses it for correctness
- c. Prints any error information to the Console window when using a Windows operating system (to `stdout` on UNIX systems) indicating the file and line where the error occurred. Possible errors include:
  - Duplicate entries
  - Ill-formed entries

---

**NOTE**

---

If you see error messages, verify the contents in the `HP700` file. Return to step 1 in “Define the Capabilities for `Computer_workstation_700`” on page 629 to correct syntax errors.

## Provide Additional Information in the `oid_to_type` File

The `oid_to_type` file is used by `netmon` to map `sysObjectIds` to the correct vendor and `SNMPAgent` values for use with `ovw` and `ovwdb`. Applications integrated with NNM use the vendor and `SNMPAgent` values to determine which menu options are active for particular devices.

### Example

This file can also be used to provide topological information to NNM. For example, use this file to automatically set all devices with this `sysObjectId` to *unmanaged* or fine-tune the manner in which NNM identifies devices with this `sysObjectId`.

The file is located in:

- *Windows:* `install_dir\conf\oid_to_type`
- *UNIX:* `$OV_CONF/oid_to_type`

There are four fields for each entry in the file:

```
sysObjectId:vendor:SNMPAgent:IP Map Topology Attributes
```

For information about controlling the vendor and SNMPAgent values, see Appendix D, “Changing an Object’s Vendor and SNMP Agent,” on page 637. For information about providing IP Map topology attributes for a specific device, refer to the comments at the top of the `oid_to_type` file and refer to the *oid\_to\_type* reference page in NNM’s online help (or the UNIX manpage).

In general, you will not need to modify the `oid_to_type` file because extensive information by `sysObjectIds` is already included.

There are actually two files that NNM uses for this purpose: `oid_to_type` and `HPoid2type`. The `HPoid2type` file contains the HP `sysObjectId` list. The important difference between these files is:

- `oid_to_type`  
If you make changes to this file, your changes will be preserved the next time you install a patch or upgrade for NNM.
- `HPoid2type`  
If you make changes to this file, your changes will be lost the next time you install a patch or upgrade for NNM.

For this example, you do not need to make any changes. Since the HP 9000 Series 700 workstation is an HP device, the `sysObjectId`: `1.3.6.1.4.1.11.2.3.2.5` is located in the `HPoid2type` file.

## Update the Database

You must now ask the topology manager to update the database, so the new symbols you created appear on the map.

1. Log in as:



- *Windows:* Administrator
  - *UNIX:* root
2. With all the NNM processes running, shut down just the netmon service. From the command prompt, type:

```
ovstop -c -v netmon
```

3. Update the database by using the `ovtopofix` command.

The `-o` option to `ovtopofix` requires the SNMP `sysObjectID`, as returned by the agent of the object you are updating. This is the same `sysObjectID` you used in the `002_HP_Other` file. From the command prompt, type:

- *Windows:*

```
ovtopofix -u -o 1.3.6.1.4.1.11.2.3.2.5 2>&1|more
```

- *UNIX:*

```
ovtopofix -u -o 1.3.6.1.4.1.11.2.3.2.5 2>&1|tee /tmp/ovtopofixOut
```

---

#### TIP

---

To avoid errors, copy and paste the `sysObjectID` from the `002_HP_Other` file to the following command line entry.

4. Restart the netmon service. From the command prompt, type:

```
ovstart -v
```

5. Check that all services are `RUNNING`. From the command prompt, type:

```
ovstatus -c
```

## Verify Symbol Changes

1. Open NNM and check for error messages:

- *Windows:*

```
Start:Programs:HP OpenView->Network Node Manager
```

**Example**

Look carefully at any messages that are output to the Console pop-up window. Messages are always appended to the end of the Console file. It is a good idea to clear the contents of this file (Edit:Clear) each time that you see it.

- *UNIX*: At the command prompt, open NNM and redirect any error messages to a file by typing:

```
ovw 2>&1 | tee /tmp/ovwOut &
```

Look carefully at any message output.

Wait for NNM to finish synchronizing—the process by which the object and map databases are made consistent. This is indicated when the [Synchronizing] message is no longer displayed in the lower left of the status bar.

2. Are all HP 700 workstations using the new symbol? To verify this example, locate all HP 700 workstations. The following steps show two ways to locate the HP 700 symbols.

- Locate the HP 700 symbol type.
  - a. From the menu bar, select Edit:Find->Object By Symbol Type.
  - b. In the Find by Type dialog box, select the computer class.
  - c. Select the HP 700 symbol subclass.
  - d. Click [Apply]. All HP 700 workstations should be listed. On the map, all of the HP 700 workstations should be represented by the new HP 700 symbol, and they should be highlighted.
- Locate objects based upon the sysObjectID attribute:
  - a. From the menu bar, select:  
Edit:Find->Object By Attribute
  - b. Scroll through the list of object attributes until you can select:  
SNMP sysObjectID.

- c. For type of string search, select `Exact Match`, and then put in the `sysObjectID` for an HP 9000 Series 700 workstation at the `Complete String` field. *Note that you must include the leading dot in the `sysObjectID` for NNM to find the object.* To locate the HP 700 workstations, enter:

```
.1.3.6.1.4.1.11.2.3.2.5
```

- d. Click `[Apply]`. All HP 700 workstations should be listed. On the map, all of the HP 700 workstations should be represented by the new HP 700 symbol, and they should be highlighted.

3. Resize a submap window that contains one of the HP 700 symbols. Does resizing the submap cause the different bitmap sizes to be used? Do the bitmaps look acceptable?

## Back Up Your Efforts

The next regularly scheduled backup will save some of the files associated with your new symbols. See “Backup/Restore to Protect Your Investment of Time” on page 149.

---

### TIP

The following *analytical data* directories are *not* included in NNM’s backup scripts:

- *Windows:*

```
install_dir\backgrounds
install_dir\bitmaps
install_dir\www\htdocs\bitmaps
install_dir\www\registration
```

- *UNIX:*

```
$OV_BACKGROUNDs
$OV_BITMAPs
$OV_WWW/htdocs/bitmaps
$OV_WWW_REG
```

If you make changes or additions to map backgrounds, bitmap files, or web registration files ensure that your new files are properly backed up.

---

Changing All the Symbols for a Particular Device

**Example**

---

# **D Changing an Object's Vendor and SNMP Agent**

## How NNM Discovers Vendor and SNMP Agent Values

This appendix describes how to modify the appropriate files to correctly set the vendor and SNMP agent fields for a particular device.

NNM automatically discovers all IP-addressable devices on your network (and IPX-addressable devices, if configured to do so on a management station running a Windows operating system). If the device has a responding SNMP agent (or proxy) that supports MIB II, then NNM will query the agent to find more information about the device. One of the pieces of information retrieved from the agent is the SNMP system object ID (`sysObjectID`) of the device:

```
iso.org.dod.internet.mgmt.mib-2.system.sysObjectID (1.3.6.1.2.1.1.2)
```

See “Unique Properties of the SNMP MIB Object `sysObjectID`” on page 447.

The discovery service then looks at the mappings defined in the `HPoid2type` and `oid_to_type` files. These ASCII files are used to set the value of the vendor and SNMP agent fields for the object.

---

### NOTE

The `HPoid2type` file contains entries that are *required* by NNM. You should *never* modify `HPoid2type`.

If no `sysObjectID` is returned by the device (that is, there is no SNMP agent running on the device) or no entry is found in these files that match the returned `sysObjectID`, the vendor and SNMP agent fields are not set. These fields are important because they can be used in the *selection rule* for certain menu items. This means that if these fields are not set, certain menu items may not be available.

These fields can also be used in `Edit:Find->Objects By Attribute`. This allows you to locate all the devices from a specific vendor. Or you can locate all the devices running a particular SNMP agent.

The vendor and SNMP agent fields for an object can be viewed from the main menu bar by selecting a symbol and `Edit:Object Properties`. Double-clicking on `General Attributes` in the `Object Attributes` list displays the `SNMPAgent` and `vendor` fields.

## When the Vendor and SNMP Agent Would Not Be Set

The vendor and SNMP agent fields for a particular object are not set if:

- There is no SNMP agent running on the device; therefore the `sysObjectID` value was never passed to NNM.
- The device's `sysObjectID` does not have a corresponding entry in either the `oid_to_type` file or the `HPoid2type` file.
- The entry in the `oid_to_type` file does not contain valid values for the vendor and SNMP agent fields of the entry.

If the `oid_to_type` file contains an entry for the `sysObjectID` but the values specified for the vendor and SNMP agent are not in the corresponding field registration files, an error will be generated when NNM is started, and the fields for that object will not be set.

The vendor and SNMP agent field values are enumerated types that are defined in the following files:

- *Windows:*  
`install_dir\fields\C\ovw_fields` and  
`install_dir\fields\C\snmp_fields`
- *UNIX:*  
`$OV_FIELDS/$LANG/ovw_fields` and  
`$OV_FIELDS/$LANG/snmp_fields`

## Procedure

This is the procedure for correctly modifying the appropriate files to set the vendor and SNMP agent fields for a particular device. These steps are explained in greater detail in the example that follows.

---

### NOTE

Most systems require `sysadmin` or `root` access to perform the following tasks.

During this procedure, you should not start any new sessions. If you do, the new sessions will use the partially built files and exit with errors. It is not necessary to shut down any NNM services. *You do need to stop and restart NNM sessions, but not until the last step in the procedure.* This minimizes loss of service.

1. Three files need to be examined and possibly modified:
  - a. `oid_to_type`  
Add or modify the entry for the `sysObjectID`.
  - b. `ovw_fields`  
Add the new vendor value.
  - c. `snmp_fields`  
Add the new SNMP agent value.
2. Reinitialize the field values.
3. Force the `netmon` service to reread the `oid_to_type` file.
4. Force the `ipmap` service to use the new `oid_to_sym_reg` mapping by closing and reopening the map.

---

### TIP

See Appendix C, “Changing All the Symbols for a Particular Device,” on page 623 for information about creating and assigning your own custom symbols to a particular `sysObjectID`.

---



## Example

The following example takes you through all the required steps in greater detail.

### oid\_to\_type

NNM identifies each device by referencing the information contained in two configuration files:

- *Windows:*

```
install_dir\conf\oid_to_type
```

```
install_dir\conf\HPoid2type (Do not edit this one.)
```

- *UNIX:*

```
$OV_CONF/oid_to_type
```

```
$OV_CONF/HPoid2type (Do not edit this one.)
```

Although the entries in these files are already extensive, you may encounter a situation where you want to make additions or changes to the `oid_to_type` file.

See the instructions at the beginning of the `oid_to_type` ASCII file. See also the `oid_to_type` reference page in NNM's online help (or the UNIX manpage) for more information.

For this example, the `sysObjectID` is: 1.3.6.1.4.1.99.1.1

In the `oid_to_type` file, use an ASCII editor to add the following entry:

```
1.3.6.1.4.1.99.1.1:NewVendor:New SNMP Agent:B
```

### ovw\_fields and snmp\_fields

The field registration files contain valid values for the object fields. The `ovw_fields` files contain the valid values for the vendor field. The `snmp_fields` files contains the valid values for the SNMP agent field.

**ovw\_fields** For this example, modify the following ASCII file to add the new vendor information:

- *Windows:*

```
install_dir\fields\c\ovw_fields
```

- *UNIX:*

```
$OV_FIELDS/$LANG/ovw_fields
```

### Procedure

`$LANG` refers to the setting of the `$LANG` language shell environment variable. To check the value of `$LANG` on your system, from the command prompt, enter:

```
echo $LANG
```

Use the value returned. If no value is returned, then use the default, which is C.

The following example illustrates how the `ovw_fields` file could be modified. Note the new entry in the file.

---

#### NOTE

These entries are case-sensitive and must exactly match the field in the `oid_to_type` file.

---

```
Field "vendor" {
Type      Enumeration;
Flags    capability, general, locate;
Enumeration "Unset",
"Hewlett-Packard",
"HP/Apollo",
"ACC",
"Cayman",
"cisco Systems",
:
"NetWare",
"NewVendor",           <-- new entry
"Novell",
"NRC",
"SGI",
"SynOptics",
"Ungermann-Bass",
"Wellfleet",
"Xyplex",
;
}
```

**snmp\_fields** For this example, modify the following ASCII file to add the new SNMP agent information:

- *Windows:*  
`install_dir\fields\c\snmp_fields`

- **UNIX:**  
\$OV\_FIELDS/\$LANG/snmp\_fields  
  
\$LANG refers to the setting of the \$LANG language shell environment variable. To check the value of \$LANG on your system, from the command prompt, enter:

**echo \$LANG**

Use the value returned. If no value is returned, then use the default, which is C.

The following example illustrates how an `snmp_fields` file could be modified. Note the new entry in the file.

---

**NOTE**

These entries are case-sensitive and must exactly match the field in the `oid_to_type` file.

---

```
Field "SNMPAgent" {
    Type      Enumeration;
    Flags     capability, general, locate;
    Enumeration "Unset",
    "HP 3000/XL",
    "HP 386",
    "HP 700/[R]X X-Terminal",
    "HP 9000/HP-UX",
    :
    "cisco Terminal Server",
    "cisco T-Router",
    "cisco Protocal Translator",
    :
    "Netware 386 TCP/IP",
    "New SNMP Agent",      <-- new entry
    "Novell Lantern",
    "NRC Fusion Xenix agent",
    "Process Software Corp. VMS agent",
    :
    "Xyplex Remote Ethernet Bridge",
    "4BSD ISODE";
}
```

### Configure the Database

It is not necessary to shut down the NNM services or to exit NNM when performing this step.

It is necessary to reinitialize the fields so that the new ones in the `snmp_fields` and `ovw_fields` files are loaded into the object database.

To reinitialize the fields, at the command prompt, type:

```
ovw -fields
```

### Updating Field Values

It is not necessary to shut down the NNM services or to exit NNM when performing this step.

To force the `netmon` service to reread the `oid_to_type` file, at the command prompt type:

```
xnmpolling -event
```

### Updating the Map

To update the map to reflect the changes, close and then reopen the map.

If the device has already been discovered, the vendor and SNMP agent changes will be detected during the next regularly scheduled configuration check. Those fields will be updated at that time to reflect the new values. You can set the intervals between configuration checks using `xnmpolling` or using the Options:Network Polling Configuration->IP menu item.

Alternately, you can delete the device's symbol from the map and let it be rediscovered. When the device is rediscovered, the new values will be used.

### Backing Up Your Efforts

The following *analytical data* directories are *not* included in NNM's backup scripts:

- *Windows:*
  - `install_dir\backgrounds`
  - `install_dir\bitmaps`
  - `install_dir\fields`
  - `install_dir\symbols`

- *UNIX operating system:*

```
$OV_BACKGROUNDS  
$OV_BITMAPS  
$OV_FIELDS  
$OV_SYMBOLS
```

When you make changes or additions to map backgrounds, bitmap files, field registration files, or symbol registration files ensure that your new files are properly backed up. See “Backup/Restore to Protect Your Investment of Time” on page 149 for more information.

Changing an Object's Vendor and SNMP Agent  
**Procedure**

---

# **E**      **Reducing NNM's DNS Lookups**

This appendix explains how NNM is designed to use a management station's IP name service more efficiently. It also explains some additional NNM configuration changes you can make to reduce the number of problem requests passed to the management station's IP name service.





## Reducing Unresolved Hostname Lookups in NNM

NNM includes functionality that automatically reduces the number of requests to a management station's IP name service:

- During NNM discovery, the `netmon` process identifies hostnames that cannot be resolved to an IP address, and adds them into a **No Lookup cache**. NNM reduces the number of repetitive failed name resolutions initiated by the event subsystem by not resolving these hostnames to an IP address.
- NNM reduces the number of repetitive hostname lookups of the host system.

NNM includes some additional tools to help you further reduce the number of hostname lookups:

- You can manually add hostnames to the `No Lookup cache` if you want to prevent NNM from initiating hostname lookups for them. See “Reducing Unresolved Hostname Lookups in NNM” on page 650 for more information.
- NNM resolves IP addresses to hostnames by initiating requests to the management station's IP name service. You can manually limit these requests by creating the `ipNoLookup.conf` file, and adding IP addresses to this file. Only add IP addresses to this file if you do not want NNM processes to initiate lookups for them using the management station's IP name service. See “Reducing IP to Hostname Lookups with the `ipNoLookup.conf` File” on page 654 for more information.

---

### NOTE

The `No Lookup cache` and the `ipNoLookup.conf` file are mutually exclusive.

NNM processes only reference the `No Lookup cache` before initiating a request to the management station's IP name service for a hostname to IP address resolution.

NNM processes only reference the `ipNoLookup.conf` file before initiating a request to the management station's IP name service for an IP address to hostname resolution.

---

- NNM caches IP addresses that cannot be resolved to a hostname by initiating a request to the management station's IP name service. NNM stores the IP address in both the IP address and hostname fields. See “Reducing IP to Hostname Lookups with Negative IP Lookup Caching” on page 655 for more information.
- NNM provides a tool that traces hostname lookups. You can use this tool to quantify the hostname lookup activity of many NNM processes. See “Using NNM's DNS Tracing Tool” on page 657 for more information.
- NNM provides a Perl script that does a quick test on the hostnames NNM is trying to resolve using the management station's IP name service. See “Using the `resolveNames.ovpl` Script” on page 662 for more information.

## Reducing Unresolved Hostname Lookups in NNM with the No Lookup Cache

Occasionally a management station's IP name service cannot resolve an object name, such as node, segment, or network name, to a valid IP address. NNM uses the No Lookup cache to store these object names, preventing multiple, and unsuccessful, requests to the IP name service.

---

### NOTE

NNM only references the No Lookup cache for hostname to IP address lookups. It does not reference the No Lookup cache for IP address to hostname lookups.

---

Many NNM processes use the SNMP cache for hostname to IP address mappings. These NNM processes query the No Lookup cache for a hostname's IP address prior to initiating a request to the management station's IP name service. If an NNM process finds a specific node in the No Lookup cache, it does not initiate a hostname lookup. This saves system resource by avoiding a failed hostname to IP address lookup.

---

### NOTE

NNM also caches IP addresses, or ranges of IP addresses, that cannot be resolved to a hostname using the management station's IP name service. See "Reducing IP to Hostname Lookups with Negative IP Lookup Caching" on page 655 for more information.

---

In the following scenarios, a management station's IP name service cannot resolve an object to an IP address:

- An IP name service cannot resolve an IP address from a hostname's MAC address. This situation occurs when a host does not have an IP address. For example, a management station's IP name service cannot resolve the following list of MAC addresses:
  - HP-861C0F
  - 3Com-921BA2
  - 0x002BB5D274C8

## Reducing Unresolved Hostname Lookups in NNM with the No Lookup Cache

- An IP name service cannot resolve an IPX address to a hostname (Windows Only). This situation occurs when a host is running IPX, and not IP. For example, a management station's IP name service cannot resolve the following IPX address:
  - 00000010:08019541710
- NNM creates a segment name when the `netmon` process discovers a new segment and assigns the segment a name. An IP name service cannot resolve a segment name to an IP address. For example, a management station's IP name service cannot resolve the following segment names, as segments are abstract objects, and are not valid hostnames:
  - MyNetwork.Segment1
  - 15.15.15.Segment20
  - 00000100.Segment1 (IPX Segments, Windows Only)

If the `netmon` process discovers a node or a segment, and determines that the name of the node or segment falls into one of the above-mentioned categories, it adds this name to the No Lookup cache.

---

**NOTE**

The `netmon` process does not add an entry to the No Lookup cache every time it fails to resolve a hostname to an IP address. A failed hostname lookup must match one of the above examples for NNM to add it to the No Lookup cache.

Once NNM makes an entry in the No Lookup cache, it makes no attempt to remove it. NNM does not assign a time-out value to the entry, therefore these entries do not automatically expire.

You can add and maintain entries to the No Lookup cache with the `snmpnolookupconf` command. See the `snmpnolookupconf` reference page (or the UNIX manpage) for more information about maintaining the No Lookup cache.

The No Lookup cache is often referred to as the `nolookupdb` database. See the `ovsnmp.conf` reference page (or the UNIX manpage) for more information about the `nolookupdb` database.

## Reducing IP to Hostname Lookups with the ipNoLookup.conf File

Not all IP addresses, or ranges of IP addresses, can be resolved to a hostname using the management station's IP name service. If NNM initiates IP to hostname lookups on these IP addresses, it can waste system resources.

To remedy this, NNM processes do not request the management station's IP name service to resolve any of the IP addresses contained in the `ipNoLookup.conf` file. You can add entries to the `ipNoLookup.conf` file to prevent NNM processes from initiating IP to hostname lookups when these situations occur.

### Configuring and Enabling the ipNoLookup.conf File

NNM does not create the `ipNoLookup.conf` file automatically. You must create and enter addresses to this file manually. To configure your management station to use the `ipNoLookup.conf` file, use the following procedure.

1. Create a file named `ipNoLookup.conf` in the following directory:
  - *Windows*: `%OV_CONF%`
  - *UNIX*: `$OV_CONF`
2. Add IP addresses and IP address ranges to the `ipNoLookup.conf` file. See the *ipNoLookup.conf* reference page (or the UNIX manpage) for further instructions on adding entries to this file.
3. Execute, as Administrator or root, `ovstop -a`.
4. Execute, as Administrator or root, `ovstart -a`.

If you need to disable the effects of the `ipNoLookup.conf` file, but retain the file and its contents, use the following procedure:

1. Execute, as Administrator or root, `ovstop -a`.
2. Rename the `ipNoLookup.conf` file.
3. Execute, as Administrator or root, `ovstart -a`.

---

## Reducing IP to Hostname Lookups with Negative IP Lookup Caching

A management station's IP name service cannot resolve all IP addresses to a hostname. When this happens, NNM caches these IP addresses in the SNMP cache, and stores each IP address in both the IP address and hostname fields. Once NNM caches this information, its processes use the IP address in place of the hostname, thereby reducing the quantity of repeated faulty lookups.

NNM's process of substituting the IP address for the hostname continues for twelve hours, or until you clear the SNMP cache. See the *xnmsnmpconf* reference page (or the UNIX manpage) for information on clearing the SNMP cache.

---

### NOTE

NNM also caches object names, such as node, segment, or network names, that it cannot resolve to a valid IP address. See "Reducing Unresolved Hostname Lookups in NNM with the No Lookup Cache" on page 652 for more information.

This feature is useful for temporarily remedying the following situations:

- This temporarily replaces the IP name service if the service becomes defective.
- This temporarily replaces a device's hostname with its IP address if NNM discovers this device prior to it being added to the IP name service.

The following shows some potential side-effects you may observe due to negative IP lookup caching.

- This may temporarily replace a fully qualified node name with its IP address.
- This temporarily replaces non-resolving node names or object selection names with IP addresses. Events containing node entries may not match these replacements.

---

**NOTE**

If the lookup problem you observe is an ongoing occurrence, refer to “Reducing Unresolved Hostname Lookups in NNM with the No Lookup Cache” on page 652 or “Reducing IP to Hostname Lookups with the ipNoLookup.conf File” on page 654 for a more permanent solution.

---



## Using NNM's DNS Tracing Tool

NNM provides a tool that traces hostname lookups. You can use this tool to quantify the hostname lookup activity of many NNM processes. When you enable DNS Tracing, each NNM process writes a separate trace file to the directory you specify when configuring the trace.

### Enabling the DNS Tracing Tool

Once you set the `OV_NS_LOG_TRACE` environment variable and restart the NNM processes, the DNS Tracing tool begins to log process activity. The DNS Tracing tool logs information according to the entries you make into the `OV_NS_LOG_TRACE` environment variable.

To enable the DNS Tracing tool for UNIX operating systems, use the following procedure:

1. As root, execute the following command:

```
OV_NS_LOG_TRACE= Trace directory;Log Threshold;Trace Mask
```

- Where *Trace directory* represents the path and name of the directory in which you want the trace files to reside. For example, if you want the logfile to go in `/tmp/dns`, then replace *Trace directory* with `/tmp/dns`.
- Where *Log Threshold* represents the time trigger point of the hostname lookups in seconds. For example if you want to log lookups greater than or equal to 600 miliseconds, then replace *Log Threshold* with `0.6`
- Where *Trace Mask* represents the level of tracing you require. The value for the Trace Mask can be a sum of any of the following trace level definitions.
  - 1: Trace the qualifying DNS queries.
  - 2: Trace the response to the qualifying DNS queries.
  - 4: Trace the qualified DN queries that were ignored, or not performed.
  - 8: Trace SNMP Cache hits for qualifying DNS queries.

- 15:Trace maximum information about DNS queries. This is the maximum value you can assign for the Trace Mask.

2. If NNM processes are running, execute, as Administrator or root, `ovstop -a`.

As Administrator, you can enable the DNS Tracing tool for Windows operating systems by using the following procedure:

1. Open the Systems applet from your Windows Control Panel and locate your Environment tab or button.
2. Add a new entry to the System Variables list.
3. Enter `OV_NS_LOG_TRACE` in the Variable field.
4. Enter the following information in the Value field:

*drive:\Trace directory,Log Threshold;Trace Mask*

- Where *drive* is the letter of the drive you want the log files to be written. For example, if you want the logfile to go in the C drive, then replace *drive* with C.
  - Where *Trace directory* represents the path and name of the directory in which you want the trace files to reside. For example, if you want the logfile to go in `\tmp\dns`, then replace *Trace directory* with `\tmp\dns`.
  - Where *Log Threshold* represents the time trigger point of the hostname lookups in seconds. For example if you want to log lookups greater than or equal to 60 seconds, then replace *Log Threshold* with `sec.60`.
  - Where *Trace Mask* represents the level of tracing you require. The value for the Trace Mask can be a sum of any of the following trace level definitions.
    - 1: Trace the qualifying DNS queries.
    - 2: Trace the response to the qualifying DNS queries.
    - 4: Trace the qualified DN queries that were ignored, or not performed.
5. Press the appropriate Set and OK buttons to set the variable.
  6. Restart your computer.

---

**NOTE**

If the NNM processes are running when you set these environment values, you must restart these processes.

- a. Execute, as Administrator or root, `ovstop -a`.
  - b. Execute, as Administrator or root, `ovstart -a`.
- 

## The DNS Tracing Tool Log Files

Once you enable the DNS Tracing Tool, NNM processes begin creating log files. The naming convention for the log files is as follows:

*processname.processid*

where *processname* represents the NNM process that is logging to the file. These processes include processes such as `netmon`, `snmpcollect`, `ovtrapd`, or others. The file extension is the process id of the process doing the logging. Each line in the trace file is a single trace message, and the parameters in each message are separated by a colon.

For example, suppose you set the `OV_NS_LOG_TRACE` environment variable as follows: `OV_NS_LOG_TRACE="c:\tmp\dns\;0.000001;7"`

The display might look something like the following text:

```
N:7:N:1003358267.712000:1003358267.727000:0.015000:"computer":Y:hostent - name =
"computer.hp.com", addresses = 17.17.17.17
A:"192.168.*.*"
A:"0-10.*.*.*"
N:7:A:1003358267.727000:1003358267.727000:0.000000:"16.16.16.16"
:Y:hostent - name = "computer1.hp.com", addresses = 16.16.16.16
L:gethostbyaddr: lookup "16.16.16.16" exceeded threshold [ 0.000001 seconds ] :
Valid response : start 1003358267.743000 - end 1003358267.759000 - difference
0.016000 seconds
I:7:A:"192.168.1.2"
I:7:N:"Segment1"
```

To understand the first parameter of this display, you need to review the following tables.

**Table E-1 Identifying the Trace Message Type**

<b>First Parameter of Trace Message</b>	<b>Message Description</b>
A	An IP address or IP wildcard has been read from the <code>ipNoLookup.conf</code> file.
E	The entry contains error information.
I	The entry contains information about an IP name service query that was ignored.
L	The entry contains information about an IP name service query that exceeded the specified threshold.
N	The entry contains information about an IP name service query attempt.

The following table describes the trace message entry values in terms of their position in the trace message. This table summarizes the most populated entry, that of a name service resolution attempt.

**Table E-2 Deciphering Log Entries**

<b>Parameter Position</b>	<b>Value Description</b>
1	Message type N for a name service query attempt.
2	Reports the trace level.
3	The type of lookup performed: A means resolution was done using <code>gethostbyaddr()</code> . N means resolution was done using <code>gethostbyname()</code> .
4	Reports the lookup start time.
5	Reports the lookup completion time.
6	Reports the amount of time required to complete the lookup.

**Table E-2 Deciphering Log Entries (Continued)**

<b>Parameter Position</b>	<b>Value Description</b>
7	Reports the value of the item looked up, such as the IP address or the hostname.
8	Did the lookup succeed? This value is either Y ( Yes) or N (No).
9	The results of the lookup.

The following table describes the trace message entry values in terms of their position in the trace message. This table summarizes IP name service queries that were ignored.

**Table E-3 Deciphering an Ignored Name Service Query**

<b>Parameter Position</b>	<b>Description of Value</b>
1	Message type I, for an IP name service query that was ignored.
2	Reports the trace level.
3	Type of lookup: A means a gethostbyaddr() call was ignored. N means a gethostbyname() call was ignored.
4	Reports the result of the IP name service query.

## Using the `resolveNames.ovpl` Script

The `resolveNames.ovpl` Perl script extracts hostnames from the output by using either the `ovtopodump` (by default) or `ovdumpevents` (`-e` option) commands. The script then attempts to resolve each hostname to an IP address using the `gethost` executable.

The script displays a list of names that could not be resolved to an IP address using the management station's IP name service. This script differentiates between names that were successfully resolved to IP addresses and names that were not resolved to an IP address. This script does not consider the amount of time required to resolve the name to an IP address.

This script can take several minutes to run depending on the size of the topology database (or the event database if the `e` option is used).

---

### NOTE

You should disable DNS Tracing `s` when using the `resolveNames.ovpl` script due to the increased number of trace files that the script will initiate.

---

The following is a usage statement for the `resolveNames.ovpl` script:

```
Usage: resolveNames.ovpl [-e] [-f file] [-i] [-n][-o  
ux10|ux11|sol|nt] [-v] [-?]
```

`-e`: get names from `ovdumpevents` (default is `ovtopodump`)

`-f`: copy output to file

`-i`: ignore IP objects (not valid with `-e` option)

`-n`: check network names (not valid with `-e` option)

`-o`: specify the operating system version

`-v`: verbose output

`-?`: print this message

**Symbols**

- .login file
  - modifying, 83
- .profile file
  - modifying, 83
- .vueprofile file
  - modifying, 83

**A**

- access
  - no access, 279
  - read-only, 278
  - read-write, 278
  - to maps, 278
- accessing
  - alarm information from the map, 326
  - map information from alarms, 325
  - MIB information, 34
- accounting management
  - components, 25
- acknowledging alarms, 321
- action block, WLRP, 486
- adding
  - background graphic, 248
  - features to NNM, 264
  - IP interface object, 246
  - new symbols, 249, 624
  - node object, 245
  - segment object, 244
- additional actions
  - alarms, 326
  - configuring, 326, 423
  - See also Alarm Browser
- additional views
  - internet view, 472
  - neighbor view, 468
  - node view, 469
  - path view, 469
  - Port-Address mapping command, 472
  - station view, 471
- address
  - resolution, 539
  - table poll operations in netmon, 164
  - well-configured, 72
- administrative symbol status, 257
- agent. See SNMP agent
- Alarm Browser
  - acknowledging alarms, 321
  - All Alarms category, 317
  - assign category, 335
  - configuring, 329
    - automatic delete, 332
    - event database, 330
    - state file, 331
  - correlated alarms, 340
  - de-duplicated alarms, 340
  - deleting alarms, 324
  - features, 502
  - filtering alarms, 322
  - overview, 315
  - starting, 503
  - troubleshooting, 582
  - window, 316, 319, 324
- alarm categories
  - configuring, 337
  - window, 315, 317
- alarm to map connection, 325
- alarms, 309, 316, 325, 405
  - accessing information, 325
  - acknowledging, 321
  - adding, 405
  - additional actions, 326
  - automatic actions, 415
  - automatic delete configuration, 332
  - categories, 335
  - categories, new, 337
  - configuration
    - sources, 414
  - configuring, 405
  - copying, 405
  - correlated
    - see event reduction capabilities
  - de-duplicated
    - see event reduction capabilities
  - definition, 309
  - deleting, 324, 405
  - deleting automatically, 332
  - displaying, 315
  - filtering, 322
  - highlighting, 326
  - launching specific URLs, 327
  - monitoring devices, 312
  - moving to a different category, 335
  - zoom to highlighted object, 325
- aliases, for web server, 475
- analytical data backup, 151
- app-default files, 255

---

# Index

Application Builder. See MIB Application Builder

application registration files (ARF)  
control menu choices, 285

archive

implement your plan, 154

importance of, 149

ARF files for menu control, 285

attributes

changing, 221

locating objects by, 96

object, 191

audit log file, 481

format, 481

location, 481

automatic actions, 415

using trusted commands, 415

automatic discovery, 38

automatic thresholds, 435

## B

background graphic

adding, 248

formats, 246

background processes

See services

backup

adding scripts, 161

archive

implement your plan, 154

importance of, 149

directory (staging area), 150

disk space requirements, 152

frequency, 151

how it works, 150

procedure, 151

SOLID database, 160

troubleshooting, 155, 158

backup routers

missing from map, 119

Berkeley Internet Name Domain (BIND)

Service, 539

bridge MIB

for topology polling, 169

bridges

discovery of, 86

display of, 203

Browse MIB operation, SNMP Set requests,  
591

browser

DMI, 399

MIB, 394

See Alarm Browser

browsing

alarms, 322

browsing MIBs

problems with, 591

See also SNMP MIB Browser

browsing MIFs. See DMI Browser

building new MIB applications, 25

bundled

correlations, 349

correlators, 371

bus layout, 228

## C

cache, clearing from browser, 576

capabilities, 192

categories for alarms

assigning, 335

changing default, 336

creating new, 337

moving an alarm, 335

CGI errors, 579

CGI programs, described, 529

change management

tools, 28

characteristics, symbol, 195

checking disk space, 511

checkpoint phases for backup, 150

child submaps

creating, 231

class, symbol, 193

client station, web components installed on,  
475

collecting

historical MIB information, 27

MIB information, problems, 570

collecting data

See data collection

collecting SNMP data. See data collection

colors, status, 257

command

xnmevents, 328

community names

configuring NNM, 75, 129

configuring SNMP agents, 75, 129

defined, 34

GET/SET, 75, 129

problems with, 589

requests for information, 34



- testing with netmon.cmstr, 75, 130
  - Community-based SNMP version 2, 34
  - Composer, 371
  - compound status propagation, 261
  - configuration check
    - operations in netmon, 164
    - polling, 168
  - configuration management
    - components, 24
    - tools, 28
  - configuring
    - additional actions, 423
    - alarm categories, 337
    - alarms, 405
    - automatic actions, 415
    - correlations, 340
    - correlators, 340
    - de-duplications, 340
    - events, 404, 405
    - polling, 163, 179
    - print command, 457
    - problems with events, 594
    - SNMP events, 405
  - connection labels
    - defined, 209
  - connection symbol, 194
  - connections, multiple connections between
    - symbols, 231
  - connector topology
    - polling, 169
  - Connector-Down correlation
    - behavior, 350
    - setting parameters, 350
      - important node filter, 177, 351
      - secondary failures, 177, 351
  - conserving memory, 620
  - container object, 234, 237
  - containment realm
    - editing guidelines, 239
    - example of Internet submap, 239
    - example of Partitioned Internet submap, 241
    - maintaining, 241
    - overview of, 238
  - context for menu items, 282
  - contrib programs
    - accessing, 56
  - controlling management traffic, 163
  - copying
    - events, 405
    - symbols, 243
  - correlating events, 340
  - correlation
    - background information, 343
    - command line control, 365
    - ConnectorDown, 350
    - files, 366
    - MgXServerDown, 354
    - new, 367
    - PairWise, 356
    - RepeatedEvent, 360
    - ScheduledMaintenance, 364
    - troubleshooting, 366
  - Correlation Composer, 371
  - correlator
    - background information, 343
    - files, 381
    - new, 382
    - OV\_Chassis\_Cisco, 372
    - OV\_Connector\_IntermittentStatus, 375
    - OV\_MultipleReboots, 373
    - OV\_NodeIf\_(group), 377
    - troubleshooting, 381
  - CPU
    - load operation, problems with, 589
    - speed, performance impact, 619
  - creating
    - child submaps, 231
    - correlations, 368
    - correlators, 382
    - de-duplication configurations, 347
    - executable symbols, 266
    - independent submaps, 231
    - Quick Navigator, 273
    - submaps, 230
  - critical status, 261
  - customizing
    - background graphic, 248
    - setting a user default map, 273
    - setting the home submap, 273
    - startup files, 520
  - cutting symbols, 243
- D**
- data collection
    - configuring, 431
    - disk space usage, 152
    - graph, 454
    - graphing, 454, 455
-

---

# Index

- reducing file size with ovdwtrend, 513
- scheduling, 433
- setting thresholds, 438
- storing results, 430
- troubleshooting, 570
- troubleshooting incorrect data, 573
- using the data collector, 429
- viewing results, 435
- Data Collector, 455
- data graphing, 455
- data link layer
  - discovery of devices, 41
- data warehouse, described, 45
- database files
  - changing permissions, 281
- databases
  - corrupted OVW object database, 566
  - event, 45
  - map, 44
  - NNM, 44
  - object, 44
  - topology, 45
  - trend, 45
- de-duplicated, 347
- deleting, 405
  - alarms, 324
  - nodes, 621
- description
  - Composer correlators, 371
  - de-duplication, 347
  - ECS correlations, 349
  - object, 222
- destination
  - SNMP trap, 311, 312
- developing
  - additional actions, 423
  - automatic actions, 415
  - de-duplication configurations, 347
  - new correlations, 368
  - new correlators, 382
- devices
  - discovery of, 87
  - supporting SNMP, 25
- DHCP
  - filter, 168
  - polling, 164, 166, 168
  - well-configured, 73
- dialog boxes
  - General Attributes, 191
- discovery, 86
  - automatic, 38
  - description of, 38
  - DIscover Level-2 Objects check box, 170
  - duplicate IP Address, 142
  - expanding, 100, 102
  - incorrect IPX node names, 140
  - IP and IPX, 40
  - IP Network 0.0.0.0, 139
  - IPX failure, 137
  - IPX node discovered as two nodes, 142
  - Level 2, 41, 170
  - limiting, 100, 107
  - MAC address, 41
  - missing IPX routers, 139
  - netmon -k options, 170
  - no vendor translation, 142
  - non-IP devices, 41
  - non-IPX devices, 41
  - partial IPX, 140
  - problems with, 137
  - recommendations, 117, 136
  - requirements for IP, 38
  - requirements for IPX, 40
  - routing tables, 127
  - start again, 144
  - unmanaging a network, 551
  - verify accuracy, 92
- discovery filter
  - procedure, 111
  - to improve performance, 515
- disk space
  - checking, 511
  - regaining, 511, 513
- displaying alarms, 316
- distributed event correlation, 346
- DMI
  - configuring, 77, 399
  - event, 36, 313
  - MIF, 36, 313
  - See MIF
  - service provider, 36, 313
- DMI Browser
  - querying DMI values, 399
- DMI devices
  - can't communicate with, 591
  - problems querying, 573, 591, 594
- DNS
  - caching-only name server, 73
  - ipNoLookup.conf file, 654

- negative IP lookup caching, 655
  - No Lookup Cache, 650, 652
  - quick check, 74
  - secondary name server, 73
  - tracing tool, 657
  - well-configured, 73, 116
  - dynamic correlation parameters, 344
  - Dynamic Views, 467
    - dvUsersManager.ovpl script, 467
    - modifying, 473
    - security, 467
  - E**
  - ECS, 349
  - e-mail message generation. See automatic actions
  - Enterprise Identification, 405
  - Enterprise MIB
    - adding, 405
    - correlating, 340
    - deleting, 405
  - environment variables
    - setting up, 83
  - error log file
    - contents, 488
    - Launcher, 616
    - using for troubleshooting, 488
    - Web Launcher, 579, 616
  - Event Browser
    - See Alarm Browser
  - Event Configuration
    - operation, 594
    - overview, 404
  - event correlation
    - bundled edition
      - Connector-Down, 177, 350
      - ManageX Server-Down, 354
      - Pair-Wise, 356
      - Repeated-Event, 360
      - Scheduled-Maintenance, 364
    - developing your own, 368, 382
    - disabling, 343
    - distributed, 346
    - enabling, 343
      - settling time, 343
    - files, 366
    - general recommendations, 345
    - introduction, 341
    - parameters
      - dynamic, 344
      - static, 344
    - See event reduction capabilities
    - sources for more, 367
    - streams, 345, 366
  - event database
    - configuration, 330
  - event database, described, 45
  - Event Identification, 405
  - event reduction capabilities
    - Correlation Composer, 371
    - de-duplication, 347
    - described, 43, 340
    - ECS, 349
    - web based, 505
  - event storm prevention
    - Composer correlators, 371
    - Connector-Down correlation, 350
    - de-duplication, 347
    - ManageX Server-Down correlation, 354
    - Pair-Wise correlation, 356
    - Repeated-Event correlation, 360
    - Scheduled-Maintenance correlation, 364
  - events, 405
    - actions, 423
    - adding, 405
    - automatic actions, 415
    - configuration, 594
      - sources, 414
    - configuring, 405
    - correlating, 340
    - de-duplication, 347
    - definition, 309
    - deleting, 405
    - processes that generate, 553
    - propagation of, 553, 566
    - services that generate, 566
  - exact match, 97
  - executable symbols
    - adding, 266
    - described, 265
  - executing MIB applications, problems, 591
  - expanding maps
    - containment realm, 238
    - creating Partitioned Internet submap, 237
    - higher level maps, 233
    - IP interface object, adding, 246
    - node object, adding, 245
    - See Also containment realm
    - segment object, adding, 244
-

---

# Index

explodable symbols  
  defined, 265

export map, 242

Extended Topology, 26, 32, 41

## F

fault management

  components, 24

  event reduction capabilities, 340

  HP OpenView IT/Operations, 27

  tools, 25

faults on network, diagnosing, 26, 28

field definitions, problems with, 558

field registration files, 629

file

  backing up originals, 540

  modifying default X resources, 541

  original product, 540

  permissions, 586

filters

  alarms, 322

  DHCP, 168

  discovery, 111, 168

  important node, 177

    Connector-Down correlation, 351

    OV\_NodeIf\_(group) correlator, 377

  map, 218

  persistence, 217

find

  attributes, 221

  entering text for exact match, 97

  entering text for pattern match, 97

  entering text for regular expressions, 97

  objects, 96, 97

forwarding

  SNMP trap from NNM, 312

  SNMP trap to NNM, 311

frequency of discovery value, in netmon, 164

FRF, 629

full restore of NNM data, 155

## G

gateways

  discovery of, 86

General Attributes dialog box, 191

generating

  alarms, 309

geometry of submaps, 252

get request, 35

get response, 35

getBulk request, 35

getNext request, 35

GIF graphics format, 246

graph

  printing, 457

  selected data, 455

Graph All Data, 454

graphics

  backgrounds for maps, 246

graphing

  data, 454

  historical data, 455

  MIB data, 27

  real-time data, 455

  with xnmgraph, 528

greyed menu items, 192

## H

help system, 49

  files, viewing, 599

  overview, 49

  printing, 50

  searching, 50

  using separately from NNM, 51

  using to create manuals, 50

hiding objects on map, 219

highlighting

  alarms, 326

  sources, 325

historical data, 455

holding area, 230

Home Base, 466

  accessing, 466

  Alarm Browser, 316

home submap

  defined, 188

  setting, 273

hotlist

  creating, 273

HP Agent

  trap recipients, 312

HP Education, 59

HP OpenView Grapher, 454

HP OpenView IT/Operations, for fault  
  management, 27

HP OpenView Launcher. See Launcher

HP OpenView Web

  applications, 474

  Correlation Composer, 371

  event correlation, 505

- features, 474
  - requirements, 475
  - security, 476
  - setting up, 475
  - web browser, 475
  - web server, 475
  - HP-RDMI service provider, 399
  - httpd service, troubleshooting, 548
- I**
- ICMP
    - polling, 164, 166
  - icon symbols
    - described, 193
  - import map, 242
  - important node filter
    - Connector-Down correlation, 351
    - OV\_NodeIf\_(group) correlator, 377
    - setting, 177
  - independent submaps, creating, 231
  - initial discovery
    - process, 90
    - start again, 144
    - troubleshooting
      - IP, 114
    - verify information, 92
  - instance
    - query for, in MIB Browser, 394
  - interface object, 95
  - Interface Traffic command
    - uses of, 617
  - Internet level, 216
  - internet view
    - description, 472
  - inventory network objects
    - internet object properties, 94
  - inventory report, 90
  - IP
    - addressing, 539
    - discovery, 38
    - forwarding, 123
    - interaction with IPX, 40
    - internet object, 94
    - objects, discovery of, 86
    - requirements for discovery, 38
    - secondary addresses, 120
    - topology behavior, 448
    - well-configured, 72
  - ipmap
    - described, 527
    - managing objects, 241
    - propagation of events from, 567
  - ipNoLookup.conf file, 650, 654
- IPX**
- discovery, 38, 39
  - interaction with IP, 40
  - layout, 39
  - polling, 164, 165, 169
  - requirements for discovery, 39
  - transport software, 82
- IPX discovery**
- duplicate IP Address, 142
  - failure, 137
  - hop count
    - expanding, 106
    - limiting, 113
  - incorrect node names, 140
  - IP Network 0.0.0.0, 139
  - missing routers, 139
  - no vendor translation, 142
  - node discovered as two nodes, 142
  - partial, 140
- J**
- Java
    - errors, 580
    - parameters, modifying, 489
  - jumpstart
    - netmon, 135
- K**
- keyboard, 456
- L**
- latency time
    - for WANs, 167
    - in polling, 167
  - Launcher
    - blank window, 580
    - CGI error, 579
    - configuring, 484
    - error log, 488, 616
    - features, 482
    - Java error, 580
    - Java parameters, 489
    - registration file. See WLRF
    - starting, 482
    - troubleshooting, 579
-

---

# Index

- user interface, 482
  - user login error, 578
  - WLRF error, 580
  - layout, 228
    - automatic, 38
    - bus, 228
    - description of, 38
    - point to point, 228
    - ring, 228
    - row/column, 228
  - layout algorithm
    - default, 227
  - level 2
    - discovery, 38, 41, 170
    - polling, 165, 169
  - license
    - checking number of nodes, 517
    - OVLICENSEMgr, 561
  - limitations, known in loading MIBs, 597
  - line
    - attributes, 457
    - configuration, 457
    - Graph, All Data, 454
    - Graph, Select Data, 455
  - list block, in WLRF, 485
  - Load/Unload MIBs operation, 596
  - loaded MIB database, modifying, 596
  - loadhost program
    - expand management domain, 105
    - limit management domain, 110
  - loading MIBs, 395
    - limitations of, 597
    - operation, 596
    - problems with, 596
    - unique object label, 597
  - local host file
    - quick check, 74
    - well-configured, 73, 116
  - Local Registration File (LRF)
    - customizing for performance, 520
    - customizing for startup, 520
  - locating objects
    - see find
  - log files
    - audit, 481
    - error, 488, 616
    - Launcher error, 616
  - logging
    - common options, 603
    - correlations, 365
    - description of, 602
    - example, 608
    - explained, 606
    - log files, 611
    - output, 612
    - recommendations, 540
    - Web Launcher errors, 579, 616
    - with netfmt, 606
    - with nettl, 607
  - lp, 457
  - LRF. See Local Registration File
- ## M
- MAC address, using to discover devices, 41
  - maintenance
    - audit log file, 481
    - daily, 511
    - error log file, 488, 616
    - monthly, 517
    - weekly, 515
  - managed object, defined, 35
  - management region
    - expanding, 91, 100, 102
    - IPX hop count, 106
    - loadhost program, 105
    - seed file, 103
  - limiting, 100, 107
    - Discovery filter, 111
    - IPX hop count, 113
    - loadhost program, 110
    - netmon.noDiscover file, 108
    - oid\_to\_type file, 112
  - troubleshooting, 114
  - management station
    - interaction with Network Presenter, 495
    - web components installed on, 475
  - manager
    - defined, 33
  - manager-agent
    - communication through SNMP, 34
  - ManageX Server-Down correlation
    - behavior, 355
    - setting parameters, 355
  - managing
    - polling, 178
  - manpages
    - accessing, 55
    - printing, 56
    - viewing, 599
-

- MANPATH statement
    - setting, 83
  - manuals, accessing online, 51
  - map database, described, 44
  - maps
    - access rights, 278
    - adding features, 264
    - automatic discovery and layout, 38
    - background graphics, 246
    - changes, 272
    - child submap, 231
    - compound status, 261
    - containment realm, 238
    - context, 282
    - copying, 201
    - creating symbols, 249
    - defined, 185
    - expanding, 233
    - file permissions, 278
    - filters for objects, 218
    - geometry, 252
    - hiding objects, 219
    - independent submap, 231
    - initial appearance, 86
    - layout
      - auto, 227
      - do-it-yourself, 228
    - limiting menu choices, 285
    - maintaining the containment realm, 241
    - menu control, 285
    - metaconnections submaps, 231
    - network symbol names, 207
    - new object holding area, 230
    - no access, 279
    - open map, 185, 201, 207
    - overlay, 253
    - partitioned internet submaps, 234
    - permissions, 278
    - persistance, 215
    - preferences, 273
    - problems with, 617
    - propagate most critical status, 262
    - propagate status at threshold value, 262
    - Quick Navigator, 274
    - read-only access, 278
    - read-write access, 278
    - remote viewing, 303
    - See also expanding maps
    - See also Partitioned Internet submap
      - size and placement, 252
      - start over, 144
      - startup options, 275
      - switch/bridge display, 203
      - troubleshooting
        - IP, 114
        - IPX, 136
      - user preferences, 273
  - marginal status, 261
  - MAU MIB
    - for topology polling, 169
  - memory
    - clearing from browser
      - , 576
    - conservation tips, 620
    - not enough, 620
    - performance impact, 619
    - problems affecting ovwdb, 566
    - related problems, 539
  - memory requirements
    - for maps, 215
  - menu items
    - adding to Network Presenter, 495
    - adding to NNM, 285
    - greyed, 192
  - menusettings.xml
    - file, 473
    - manpage, 473
  - meshing
    - definition of, 42
    - example, 213
    - specifying in NNM, 214
  - meta-connection
    - submap, 232
    - symbol, 231
  - MIB, 25
    - accessing information, 34
    - Application Builder operation, 425, 591, 592
    - bridge
      - for topology polling, 169
    - Browse operation, 28
    - building new applications, 25
    - collecting historical information, 27
    - data, 454
    - data collection
      - effects of, 435, 617
      - operation, 570
    - defining thresholds, 26
    - operation for, 435
    - DMI conversion from MIF, 77, 401
-

---

# Index

- enterprise-specific objects, 25
- graphing data, 27
- loading (adding), 395
  - prerequisites, 396
- MAU
  - for topology polling, 169
- numeric values, 429, 437
- object, 25
- object label, unique, 597
- obtaining MIB files, 76
- querying, 394
- repeater
  - for topology polling, 169
- See Also data collection
- setting MIB values, 394
- setting values, 394
- SNMP traps, 26
- storing data for trend analysis, 27
- variable bindings, 387
- vendor-specific, 395
- viewing, 394
- viewing MIB values, 394
- MIB Application Builder
  - and DMI device, 591, 594
  - using, 425
- MIB applications
  - executing, 592
  - problems building, 591, 592
  - troubleshooting, 592
- MIB Browser. See SNMP MIB Browser
- MIB event thresholds, defining, 26
- mib.coerce file, 446
- mibExpr.conf file, 445
- Microsoft Terminal Services, 303, 304
- MIF, 36, 313
  - configuring for NNM, 77, 401
  - loading (adding), 77, 401
    - with translation to MIB, 402
    - without translation to MIB, 402
  - mapping to MIB, 314
  - vendor-specific, 402
- modifying
  - events, 405
  - Java parameters, 489
  - line attributes, 457
  - ovsuf, 520
- monitoring
  - devices, 312
- mouse and keyboard
  - grapher, 456
- moving
  - symbols, 243
- multi-homed node, discovery of, 86
- N**
- name resolution, 73, 74, 116
- naming network symbols, 207
- negative IP lookup caching, 655
- neighbor view, 468
- NetBios
  - quick check, 74
  - well-configured, 73, 116
- netfmt, 606
  - described, 606
  - examples of, 612
  - filtering options, 612
- netmon, 448
  - address table poll operations, 164
  - background service, 548
  - configuration check operations, 164
  - configuring, 178
  - dependencies, 546
  - described, 525, 548
  - disabling, 180
  - discovery behavior, 525
  - enabling, 180
  - event generation conditions, 566
  - frequency of discovery values, 163
  - instance restrictions, 551
  - IP discovery, 38
  - IPX discovery, 39
  - jumpstart(-J), 135
  - monitoring, 179
  - netmon.noDiscover file, 108
  - performance check, 178
  - permissions, 586
  - poll operations, 164
  - problems with, 137
  - proxies, 164
  - queue (-q, -Q), 178
  - requirements for discovery, 38, 39
  - seed file, 102
  - timeout intervals, 164
  - trace masks, 604
  - tracing, 604
  - tracing example, 604
  - troubleshooting, 548
    - tools, 551
- netmon.cmstr, 130



- testing community names, 75
  - netmon.equivPorts, 42, 212, 214
  - netmon.lock file, 551
  - netmon.lrf, 178
    - described, 520
    - setting segRedux, 204
  - netmon.noDiscover file
    - effect on discovery, 109
    - example, 109
    - HSRP virtual IP address, 108
    - procedure, 109
  - netmon.snmpStatus, 260
  - netnmc script
    - execution sequence, 588
  - nettl
    - described, 606
    - examples of, 608
    - generating log files, 611
    - generating trace files, 611
    - log file output, 612
    - options
      - log, 608
      - start, 607
      - status, 607
      - stop, 611
      - traceoff, 610
      - traceon, 609
    - pathnames for, 606
    - subsystems used with, 606
  - network level, 216
  - network management
    - benefits, 24
    - defined, 24
    - functions, 24
    - proactive, 22
  - network management operation
    - problems with, 589
    - remote node support for, 589
  - network map
    - using a seed file, 102
  - network object, 94
  - Network Presenter
    - components, 492
    - connecting to an NNM session, 492
    - features, 491
    - features vs. NNM, 493
    - interaction with management station, 495
    - permissions, 582
    - problems starting, 581
    - registration file. See NPRF
      - specifying map to open, 492
      - starting, 491
      - symbol registration file, 496
      - troubleshooting, 581
      - URL for, 491
  - network traffic
    - control, 163
  - NetworkAdmin user role, 478
  - NetworkOper user role, 478
  - networks
    - naming symbols, 207
  - new node discovery
    - polling, 170
  - NIS
    - quick check, 74
    - well-configured, 73, 116
  - NNM
    - configuration files, changing permissions, 281
    - options for starting, 275
    - starting, 89, 104, 105, 180
    - status check, 89
  - NNM Advanced Edition, 32, 41
    - Alarm Categories, 317
  - NNM Extended Topology, 26
  - NNM Starter Edition
    - Alarm Categories, 317
  - no access, 279
  - No Lookup Cache, 650, 652
  - node
    - deleting unwanted, 621
    - discovery of IP, 39
    - discovery of IPX, 39
    - information obtainable from, 542
    - information storage, 39
    - multi-homed, 86
    - non-SNMP, 542, 589
    - object, 95
    - represented as a symbol, 88, 641
  - node view, 469
    - description, 469
  - normal status, 261
  - notification, 311
    - defined, 34
  - NPRF
    - adding menus, 495
    - adding toolbar buttons, 495
    - format, 495
    - location, 495
    - troubleshooting, 582
-

---

# Index

NTAdmin user role, 478  
NTOper user role, 478  
numeric MIB values, 429, 437

## O

object database  
  and ovwdb, 526  
  described, 44

object manager  
  ovstatus cannot find, 545

### objects

  adding attributes, 224  
  attribute, 191  
  capabilities, 192  
  changing attribute sets, 223  
  changing attribute values, 222  
  container, 234  
  context, 282  
  defined, 34, 190  
  filtering on maps, 218  
  general attributes, 191  
  getting description, 222  
  hiding on map, 219  
  holding area, 230  
  interface, viewing description of, 95  
  IP interface, adding, 246  
  IP internet, viewing description of, 94  
  keep in memory, 216  
  locating by, 96, 97  
  managed, 35  
  network, viewing description of, 94  
  node, adding, 245  
  node, viewing description of, 95  
  segment, adding, 244  
  segment, viewing description of, 95  
  SNMP agent attribute, 638  
  status, 256  
    propagate, 260  
  unmanaged, 35  
  vendor attribute, 638  
  when to manage, 36  
oid\_to\_sym\_reg, 88, 122, 125, 224, 251, 624, 629  
  changing a symbol, 624, 628  
  changing SNMP agents, 640  
  oid\_to\_type, 625  
oid\_to\_type, 88, 122, 125  
  changing a symbol, 624, 631  
  changing SNMP agents, 640

  unmanaging devices, 112, 178  
  update/verify changes, 632  
on-demand submap, to conserve memory, 621  
online help  
  described, 49  
  menu, 49  
  printing, 50  
  searching, 50  
  using separately from NNM, 51  
  using to create manuals, 50  
online manuals, accessing, 51  
open map  
  defined, 201  
  defined, 185, 207  
OpenView Forum, 59  
operation, network  
  problems with, 586  
  troubleshooting, 586  
operational data, backup, 150  
operational symbol status, 257  
operator actions, 423  
OV\_Chassis\_Cisco correlator, 372  
  behavior, 372  
  setting parameters, 372  
OV\_Connector\_IntermittentStatus  
  correlator, 375  
  behavior, 375  
  setting parameters, 375  
OV\_MultipleReboots correlator, 373  
  behavior, 373  
  parameter settings, 374  
OV\_NodeIf (group) correlator, 377  
  behavior, 377  
  disabling, 379  
  important node filter, 377  
  setting parameters, 378  
ovactiond  
  dependencies, 546  
  described, 525  
  troubleshooting, 552  
  trusted commands, 415  
  trustedCmds.conf directory, 415, 416  
ovactiond.lrf, described, 521  
ovaddobj command  
  described, 525  
  netmon, configuring seed file, 108  
OVAdmin user role, 479  
ovalarmadm, 334  
ovalarmsrv  
  dependencies, 546  
  described, 526

- troubleshooting, 552
- ovas
  - described, 526
- ovbackup.ovpl, 151, 154
- ovcapsd, 36, 313, 402
  - dependencies, 546
  - described, 526, 553
  - troubleshooting, 553
- ovcapsd.lrf, described, 521
- overlay of submaps, 253
- overview
  - Alarm Browser, 315
  - Event Configuration, 404
- ovhtpasswd command, 477
- ovrepld
  - dependencies, 546
  - described, 526
  - troubleshooting, 556
- ovrepld.lrf, described, 521
- ovrequestd, 526
- ovrequestd background service, 569
- ovrestore.ovpl, 156, 157
- ovspmd
  - described, 525
  - determining problems, 554
  - recreating sockets directory, 555
  - troubleshooting, 554
- ovstart command
  - described, 525
  - in starting background services, 544
- ovstatus command
  - failure to find object manager, 545
  - verifying daemon services, 545
- ovsuf
  - described, 525
  - modifying, 520
- ovtopmd
  - dependencies, 546
  - described, 526, 556
  - troubleshooting, 556, 557
- ovtopmd.lrf, described, 521
- ovtopodump
  - description of, 559
  - troubleshooting, 560
- ovtopofix command, 206
- ovtrapd
  - dependencies, 546
  - described, 526, 561
  - operation of, 561
  - troubleshooting, 561
- ovtrapd.lrf, described, 521
- ovuispmd
  - dependencies, 546
  - described, 526, 562
  - troubleshooting, 562
- ovuispmd.lrf, described, 521
- ovw command
  - described, 527
  - options, 275
    - mapcount, 275
    - verify, 275, 289
- ovw.auth file, location, 496
- ovwchgrp, 281, 587
- ovwchmod, 281, 587
- ovwchown, 281, 587
- ovwdb
  - corrupted OVW object database, 566
  - dependencies, 546
  - described, 526, 564
  - insufficient memory, 565
  - insufficient performance, 565
  - troubleshooting, 564
- ovwdb.auth file
  - location, 496
- ovwdb.lrf, described, 521
- ovwls, 281, 587
- ovwperms, 281

**P**

- paging automatically. See automatic actions
- Pair-Wise correlation
  - behavior, 357
  - setting parameters, 358
- palette, 193
- Partitioned Internet submap
  - creating, 237
  - example of, 241
  - how it is managed, 241
  - overview of, 234
  - See Also containment realm
- password
  - in user authentication file, 477
  - modifying, 477
- pastng symbols, 243
- patch releases, verifying, 517
- PATH statement
  - modifying, 83
- path view, 469
- pause NNM services
  - for back up
    - analytical, 151, 152

---

# Index

- operational, 152
- Performance
  - Graph Data, 454
- performance, 455
  - customizing startup, 520
  - diagnosing problems, 26, 28
  - improving NNM, 621
  - increasing, 619
  - limiting factors, 619
  - polling, 515
  - problems with, 619
- performance management
  - components, 24
  - reports, 27
  - tools, 27
- Perl (HP's version)
  - backup/archive/restore, 161
- permissions
  - change for configuration files, 281
  - change for database files, 281
  - changing, 587
  - default, 586
  - no access, 279
  - read-only, 278
  - read-write, 278
  - restoring default, 587
- persistance of submaps, 215
- persistence filter, 217
- planning, 62
  - assuming responsibility, 64
  - changes to the map, 272
  - how to configure NNM, 65
  - which devices to manage, 69
- pmd, 309
  - alarm gathering, 310
  - dependencies, 546
  - described, 526, 566
  - operation of, 566
  - permissions, 586
  - troubleshooting, 366, 381, 553, 566
- pmd.lrf, described, 522
- point to point layout, 228
- poll operations, in netmon, 164
- polling
  - adjusting intervals, 515
  - from a remote computer, 167
  - behavior, 551
  - causing problems, 540
  - configuration check, 165, 168
  - controls, 163, 550
  - delays in, 551
  - disabling, 180
  - during map generation, 163
  - effects of reducing, 163, 164, 617
  - file, 550
  - initial process, 38
  - latency time, 167
  - new node discovery, 163, 165, 170
  - non-IP devices, 41
  - non-IPX devices, 41
  - performance
    - checking, 515
  - problems with, 551
  - secondary failures, 165, 177
  - secondary IP addresses, 120
  - status, 163, 164, 166
  - to conserve memory, 620
  - topology check, 165, 169
  - traffic generated by, 617
  - troubleshooting
    - IP, 114
    - jumpstart netmon, 135
    - SNMP, 75, 128
    - subnet masks, 73, 132
  - tuning, 178, 179
  - values storage, 163, 527, 550
- Port Address Mapping command, 472
  - description, 472
- port number, used to start HP OpenView Web, 475
- port trunking
  - definition of, 41
  - example, 211
  - specifying in NNM, 212
- ports, connection problem, 558, 565
- predefined symbols, 194
- preferences, 273
- preferences, setting a user default map, 273
- print command, 457
  - configuration, 457
- printing
  - graph, 457
  - online help, 50
- proactive network management, 22
- problem management
  - components, 24
  - tools, 25
- problems, 114
  - additional sources of information, 539
  - address resolution, 539

- analyzing symptoms, 542
- background services, 544
- Browse MIB operation, 591
- characterization, 542
- diagnosing network faults, 26, 28
- diagnosing performance, 26, 28
- event correlation, 366, 381
- failure to discover IPX nodes, 137
- Launcher won't start, 580
- launching applications, 600
- limitations on loading MIBs, 597
- loading MIBs, 596
- map, 617
- netmon, 137, 551
- network management operation, 589
- network operation, 586
- ovtopmd, 557
- ovtopodump, 560
- ovwdb, 565
- performance, 619
- pmd, 568
- polling, 551
- resource exhaustion, 539
- runtime, 586, 588
- starting Launcher, 576
- starting Network Presenter, 581
- subnet mask, 132
- unmanaging a network, 551
- user logins, 578
- X Windows, 598
- product support
  - contacting, 59
- propagating symbol status, 257
  - most critical status, 262
  - status at threshold value, 262

## Q

- quick check
  - name resolution, 74
- Quick Navigator, 274
  - creating, 273

## R

- read-only
  - access, 278
- read-write
  - access, 278
- real-time data, 455
- real-time data graphing, stopping, 456

- recreating map and topology databases, 144
- reference pages
  - accessing, 54
  - printing, 54
  - viewing, 599
- registering an application, 264
- registration files
  - field, 629
  - Launcher. See WLRF
  - Network Presenter. See NPRF
  - symbol, 626
  - update/verify, 630
- regverify command, 484
- release notes, accessing, 53
- remote access to NNM, 303
- Remote Ping
  - problems with, 589
- remoteConfAllow.conf file, 167
- Repeated-Events correlation
  - behavior, 361
  - setting parameters, 361
- repeater
  - discovery of, 86
- repeater MIB
  - for topology polling, 169
- report
  - configuration interface, 27
  - general availability, 27
  - general inventory, 27
  - inventory, 90
  - modification, 27
- Reporting interface, 506
  - troubleshooting, 577
- reports
  - removing, 513
- restarting automatic map generation, 144
- restore
  - adding scripts, 161
  - all NNM data, 155
  - some NNM data, 157
- ring layout, 228
- root submap, 188
- routing tables
  - in IP discovery, 127
- row/column layout, 228
- running services, checking, 511
- runtime components on network
  - problems with, 586

---

# Index

## S

Scheduled-Maintenance correlation  
behavior, 364

setting parameters, 364

searching online help, 50

secondary failures

Connector-Down correlation, 351

OV\_NodeIf\_(group) correlator, 377

polling, 177

security

audit log file, 481

for HP OpenView Web, 476

ovw.auth file, 496

ovwdb.auth file, 496

session configuration file, 480

user authentication file, 477

user roles file, 478

security management

components, 25

seed file

description of, 103

effect, 105

example, 103

format, 109

location, 109

name, 109

networks implied by, 109

when to use, 102

segment

object, 95

segment level, 216

Selected Nodes, 455

selecting

sources that correspond with alarms, 326

selection name, locate object by, 96

service provider

See DMI

services

associated port numbers, 546

at start-up time, 531

checking running, 511

dependencies on other services, 545

during operation, 532

logging and tracing, 602

netmon, 448

pmd, 309

starting with ovstart, 544

troubleshooting background, 544

verifying with ovstatus, 545

web interface, 529

session configuration file, 480

defaults, 480

format, 480

location, 480

set operation, 34

set request, 35

setting

community names, 128

filters, 322

home submap, 273

MIB values, directly, 394

MIB values, using query, 394

user default map, 273

setting your path, 83

settling time

event correlation, 343

Snapshot Recover operation, 118

SNMP, 455

agents required, 117

community names, 129

configuration, 128, 129

data graph, 454

data graphing, 455

event configuration, 405

event correlation, 340

nodes without, 41, 542, 589

operations for accessing MIB information,  
34

polling, 163, 456

port, 129

proxie for agent, 129

SetRequests in a Browse MIB operation,  
591

traps, 26

variable bindings, 387

SNMP agent

changing, 640

defined, 34

management operations exclusive to, 589

non-HP, 589

propagation of events from, 561

required for SNMP, 117

setting attribute, 638

unknown, 131

well-configured

community names, 75

MIB files, 75

trap forwarding, 75

SNMP Data Presenter

configuring, 501

- features, 499
- starting, 500
- SNMP MIB Browser
  - querying, 394
  - setting values, 394
  - troubleshooting SNMP, 119, 123, 130
  - viewing, 394
- snmp notify, 311
- SNMP services, 35
  - get request, 35
  - get response, 35
  - getBulk, 35
  - getNext, 35
  - set request, 35
- SNMP traps, 312
  - forwarding from NNM, 312
  - forwarding to NNM, 311
- snmpCol.trace file, 571
- snmpColDump, 570
- snmpCollect
  - dependencies, 546
  - described, 527
  - tracing, 572
  - troubleshooting, 570
- snmpCollect.lrf, described, 522
- snmpd
  - errors, 598
  - propagation of events from, 561
- SNMPv1
  - traps, 311
- SNMPv2c, 311
  - traps, 311
- sockets
  - recreating ovsmppd sockets directory, 555
- SOLID database
  - discreet backup, 160
- sources that correspond with alarms, 326
- sparse files (UNIX only)
  - backup, 155, 156, 157
- special variables, 418
- SRF, 626
- staging area for backups, 149
- standard deviation, 435
- star
  - layout, 228
- starting NNM, 89, 90, 104, 105, 144, 157, 158, 180
  - redo initial discovery, 144
- startup
  - automatic
    - disabling, 519
    - enabling, 519
    - customizing for performance, 520
  - state file
    - configuration, 331
  - static correlation parameters, 344
  - station view
    - description, 471
  - status
    - categories, 257
    - colors, 257
    - compound source, 259
    - compound, explained, 261
    - conditions, 257
    - critical, 261
    - devices outside firewall, 169
    - marginal, 261
    - normal, 261
    - object source, 259
    - polling, 166
    - polling by object class, 167
    - polling with SNMP, 168, 259
    - propagate most critical, 262
    - symbol status, 257
    - unknown, 261
  - stopping
    - all NNM sessions, 305
    - NNM to redo discovery, 144
    - real-time data graphing, 456
  - storing, MIB data for trend analysis, 27
  - streams and event correlation, 345, 366
  - subclass, symbol, 193
  - submaps, 231
    - background graphic, 188, 246, 248
    - creating, 230, 231
    - defined, 186
    - geometry, 252
    - home, 188
    - independent, creating, 231
    - layout algorithm, 227
    - meta-connection, 232
    - Node, 245
    - overlay, 253
    - Partitioned Internet, 234
    - persistence, 215
    - Quick Navigator, 274
    - regular vs. meta-connection, 232
    - root, 188
    - Segment, 245
    - setting, 273
  - subnet masks, 133

---

# Index

- hierarchical, 73, 132
  - non-contiguous, 73, 132
  - problems with, 132
  - troubleshooting, 73, 132
  - subsystems used with nettl command, 606
  - support
    - obtaining, 59
    - programs provided, 57
  - swap space related problems, 539
  - switches
    - display of, 203
  - symbol
    - See node
  - symbol registration file, 626
    - for Network Presenter, 496
  - symbols
    - adding, 244
    - behavior, 265
    - changing, 624
    - characteristics, 195
    - class, 193
    - color
      - propagate, 260
      - status, 256
    - connection, 194, 232
    - context, 282
    - copying, 243
    - creating, 249, 624
    - cutting and pasting, 243
    - defined, 193
    - executable, 265
    - explodable, 265
    - hiding on map, 219
    - icon, 193
    - meta-connection, 231
    - moving, 243
    - multiple connections between, 231
    - naming networks, 207
    - predefined, 194
    - See node
    - status, 96, 257
    - subclass, 193
    - type, 97, 193
    - varieties, 193
  - synchronized data
    - restore all NNM data, 155
  - sysObjectID, 638
    - definition, 447
    - describing an SNMP agent's source, 448
    - determining IP topology behavior, 448
    - determining symbol type, 448
    - event reduction, 340
    - identifying an event's source, 448
    - netmon, 448
    - system registry, 254
      - map access, 289
      - map appearance, 254
- ## T
- tab block, WLRP, 485
  - thresholds
    - file, 550
    - operation for defining, 435
  - time intervals, 455
  - timeout intervals, 164
  - Tip of the Day
    - behavior, 51
  - toolbar buttons
    - adding to Network Presenter, 495
  - topology
    - polling, 169
  - topology database
    - and ovtopmd, 526
    - described, 45
  - total restore of NNM data, 155
  - trace file
    - trimming, 513
  - tracing
    - cautions for using, 602
    - common options, 603
    - data collection, 572
    - description of, 602
    - netmon, 604
    - recommendations, 540
    - trace files, 611
    - trimming file, 513
    - turning on, 610
    - with nettl, 609
  - traffic generation
    - measurement of polling, 617
    - regulation of polling, 163, 617
  - training
    - HP Education, 59
  - trapd.conf, 404, 414
    - file, 594
  - trapd.log, 330
  - traps, 311, 404
    - defined, 34, 311
    - forwarding from NNM, 312
    - forwarding to NNM, 75, 311
-



- recipients, 312
- SNMP, 26
- trend database, described, 45
- trimming
  - trace file, 513
- troubleshooting, 114
  - additional sources of information, 539
  - Alarm Browser, 582
  - analyzing problems, 542
  - archive, 155, 158
  - background services, 544
  - backup, 155, 158
  - collecting too much data, 540
  - defaults modifications, 541
  - DMI communication, 591
  - IP, 114
  - Launcher, 579
  - Launcher error log file, 488, 616
  - name resolution scheme, 73, 74, 116
  - network operation, 586
  - Network Presenter, 581
  - OVLICENSEMgr, 561
  - polling too much, 540
  - recommended practices, 539
  - restart initial discovery, 144
  - SNMP agents, 129
  - symptoms, 542
  - web components, 576
- trustedCmds.conf directory, 415, 416

## U

- universal pathnames
  - for UNIX, 82
  - for Windows, 84
  - setting up, 83
- UNIXAdmin user role, 479
- UNIXOper user role, 479
- unknown status, 261
- unmanage objects, 100
  - automatically, 112, 178, 631
  - defined, 35
  - effects of, 178, 617
  - uses of, 178, 617
- user authentication file, 477
  - location, 477
  - modifying, 477
- user default map
  - setting, 273
- user preferences, 273

- user roles
  - described, 478
  - NetworkAdmin, 478
  - NetworkOper, 478
  - NTAdmin, 478
  - NTOper, 478
  - OAdmin, 479
  - UNIXAdmin, 479
  - UNIXOper, 479
- user roles file, 478
  - format, 479
  - location, 479

## V

- values
  - MIB, 394
- variable binding
  - event reduction, 340
  - how to find and use, 387
- variables, 418
- vendor attribute, 638
- viewing
  - connection labels, 209
  - events, 316
  - MIB values, 394
  - object descriptions, 94, 95
  - see find

## W

- WANs
  - latency time for, 167
- web browser
  - requirement on client station, 475
- web interface services, 529, 534
- web server
  - aliases, 475
  - included in NNM, 475
- web sites
  - for NNM manuals, 58
  - HP OpenView, 60
  - HP Openview, 58
  - OpenView Forum, 59
  - product support, 59
- WebWindow, 486
- well-configured network
  - community names, 75
  - DHCP, 73
  - DMI service providers, 77
  - MIB files obtained, 76

---

# Index

- name resolution, 73, 116, 124, 126
- SNMP, 128
- subnet masks, 73, 132
- trap forwarding, 75
- troubleshooting, 114
- white papers, accessing, 57
- wildcards
  - event sources, 414
  - netmon.noDiscover file, 109
  - threshold monitoring, 434
- window, 317, 319
  - Alarm Browser, 315
  - Alarm Categories, 315
- window geometry, 252
- Windows
  - system registry, 254
- WLRF, 484
  - action block, 486
  - components, 484
  - list block, 485
  - location, 484
  - problem reading, 580
  - tab block, 485
  - verifying, 484
  - Web Launcher Registration File, 484
  - WebWindow, 486
- www access, 304

## X

- X Windows
  - .Xdefaults file, 541
  - performance impact, 619
  - problems with, 598
  - resources, 541, 598
  - restoring default resources, 598
- X11 monochrome bitmap format, 246
- xnm program
  - operations started by, 589
  - permissions, 586
- xnmappmon
  - described, 527
  - operations started by, 590
- xnmbrowser, described, 527
- xnmbuilder, described, 527
- xnmcollect, described, 527
- xnmevents
  - command, 328
- xnmeventsExt.conf file, 327
  - launching views and URLs, 327
  - modifying, 327

- xnmgraph, described, 528
- xnmloadmib, described, 528
- xnmpolling, 179
- xnmpolling, described, 528
- xnmsnmpconf, described, 528
- xnmtrap, 404
- xnmtrap, described, 528
- xpr, 457
- xwd, 457

## Z

- zoom, 456



