Managing NerveCenter

UNIX and Windows
Version 5.1.06

LogMatrix, Inc.
4 Mount Royal Ave, Suite 250
Marlborough, MA 01752
Toll Free +1 (800) 892-3646
Phone +1 (508) 597-5300
Fax +1 (774) 348-4953
info@logmatrix.com
http://www.logmatrix.com

Contents

1 Introduction

Overview of this Book 2
NerveCenter Documentation 4
Using the Online Help 4
Printing the Documentation 4
The NerveCenter Documentation Library 5
UNIX Systems 6
Document Conventions 6
Documentation Feedback 7
LogMatrix Technical Support 7
Professional Services 7
Educational Services 8
Contacting the Customer Support Center 8
For Telephone Support 8
For E-mail Support 8
For Electronic Support 8
For Online KnowledgeBase Access 8
For User Community Access 9

2 Understanding NerveCenter

NerveCenter Overview 12
How NerveCenter Correlates Events 13
How NerveCenter Manages Nodes 14
How Behavior Models Work 15
Behavior Model Objects 15
Behavior Model Operation 16
How Properties Affect Behavior Models 17
Other Trigger Generators 17
Alarm Actions 18
Alarms 18
NerveCenter Components 21
NerveCenter Server 21

NerveCenter Database 21
NerveCenter MIB 22
NerveCenter Interfaces 23
The NerveCenter Administrator 24
The NerveCenter Client Console 25
The NerveCenter Client 26
The NerveCenter Web Client 27
The Command Line Interface 28
Role in Network Management Strategy 29
Standalone Operation 30
Using Multiple NerveCenter Servers 31
Integration with Network Management Platforms 32

3 Administering NerveCenter

Who Uses NerveCenter? 34
NerveCenter Login Rights 35
The Role of a NerveCenter Administrator 36

4 Running the NerveCenter Server

Running the NerveCenter Server on UNIX 38
Running the NerveCenter Server on Windows 39
Troubleshooting: Running the NerveCenter Server 40

5 Connecting to a NerveCenter Server

Starting NerveCenter Administrator 42
Starting the NerveCenter Administrator on UNIX 42
Starting the NerveCenter Administrator on Windows 42
Connecting Administrator to a NerveCenter Server 43
Connecting Administrator to Multiple NerveCenter Servers 44
Deleting a Name from the Administrator Server List 45
Viewing NerveCenter Server Status 46
Disconnecting the Administrator from a NerveCenter Server 49
Troubleshooting: Connecting to the NerveCenter Server 50

6 Managing the NerveCenter Server

Managing the NerveCenter Trap Source 52
Trap Source Overview 52

NerveCenter Trap Utilities 53
Trapgen 53
Traprcv 54
Changing the SNMP Trap Source 55
Reconfiguring a NerveCenter Server from the Command Line 61
Troubleshooting: Managing the NerveCenter Server 71

7 Managing NerveCenter Port Settings

Client and Server Communication Ports 74
Configuring the NerveCenter Server Connection Port. 74
Changing the NerveCenter Administrator Server Port. 76
Changing the NerveCenter Client Server Port 77
Configuring NerveCenter to Receive Inform Actions. 78
Changing the Command Line Interface Port. 80
Specifying SNMP Ports for NerveCenter 81
Troubleshooting: Managing NerveCenter Port Connections 83

8 Managing Node Data

How NerveCenter Manages Nodes 86
Defining a Set of Nodes 86
Detecting Conditions 87
Correlating Conditions 87
Detecting the Persistence of a Condition 88
Finding a Set of Conditions. 89
Looking for a Sequence of Conditions 90
Responding to Conditions 92
Notification 93
Logging 93
Causing State Transitions 94
Corrective Actions. 94
Action Router 95
The NerveCenter Node List. 96
Filtering Nodes 97
Filtering Using Node Capabilities 97
Filtering Using a Node System Object Identifier. 98
Filtering Nodes by IP Address 99
Filtering Nodes by Hostnames 104
Enabling and Disabling IP and Hostname Filters. 105
Processing Traps from Unknown Nodes 107

Initially Populating the Node List. 109
    Populating Using a Network Management Platform as a Data Source 110
    Populating Using the IPSweep Behavior Model 112
    Populating the Node List Manually. 114
Maintaining the Node List 115
    Synchronization with Your Network Management Platform 115
    Adding Nodes Discovered from Traps 117
    Adding and Deleting Nodes Manually 119
Troubleshooting: Managing Node Data 121

9 Managing SNMP Settings

Overview of NerveCenter SNMPv3 Support 126
    NerveCenter Support for SNMPv3 Security 128
    NerveCenter Support for SNMPv3 Digest Keys and Passwords 129
The Need for Node Classification 130
    The NerveCenterNode Classification Algorithm 131
    How NerveCenter Classifies a Node 132
SNMP Auto and Manual Classification Settings 133
    How NerveCenter Classifies Node SNMP Versions. 133
    When NerveCenter Classifies Node SNMP Versions. 134
    Classifying Nodes Automatically 135
    Setting a Maximum Classify Value. 137
Performance Tuning for Node Classification 138
    Setting the Maximum SNMPv3 Requests per Cycle. 138
    Specifying SNMP Poll Intervals for NerveCenter. 139
    Multi-Threaded Polling 140
Specifying NerveCenter SNMP Ports 143
Discovery and Initialization of SNMPv3 Agents 144
Configuring an SNMPv3 Agent for NerveCenter 145
Configuring an Initial User for Discovering an SNMPv3 Agent 146
Configuring SNMPv3 Security Settings 147
Viewing the SNMPv3 Operations Log. 149
    Signing a Log for SNMPv3 Errors Associated with Your Administrator. 150
    Signing a Log for SNMPv3 Errors Associated with a Remote Client or Administrator. . . . 151
    Viewing the SNMPv3 Operations Log 152
SNMP Error Status 153
SNMP Agent. 157

10 Managing ICMP Settings

Overview of ICMP Support . . . 161
NerveCenter Support for ICMP Polling . . . 161
NerveCenter Support for ICMP Event Monitoring . . . 162
Configuring NerveCenter for ICMP Event Monitoring . . . 162

11 Managing NerveCenter Security

Managing Security on UNIX . . . 166
Troubleshooting UNIX Login Issues . . . 168
Managing Security on Windows . . . 170
Adding Users to a NerveCenter on a Windows Workstation or Server . . . 170
Adding Users to a NerveCenter on a Windows Domain Controller . . . 172
Working with Active Directory . . . 174
Troubleshooting: Managing NerveCenter Security . . . 176

12 Managing NerveCenter Alarm Actions

Specifying an SMTP Server for Mail Notification . . . 180
Configuring NerveCenter to Send Microsoft Mail . . . 182
Configuring NerveCenter to Send Microsoft Mail While Running as a Service . . . 183
Configuring NerveCenter to Send Microsoft Mail while Running the Server as a Process . . . 186
Setting Dialing Properties for Paging Notification on Windows . . . 188
Setting Dialing Properties for Paging Notification on UNIX . . . 189
Specifying Settings for Log Management . . . 191
Troubleshooting: Managing Alarm Actions . . . 194

13 Managing NerveCenter Web Integration

How to Start and Stop the NerveCenter Web Collector . . . 196
Starting and Stopping the NerveCenter Web Collector in UNIX . . . 196
Starting and Stopping the NerveCenter Web Collector in Windows . . . 196
Configuring a Web Server for NerveCenter . . . 197
Configuring Internet Information Server . . . 198
Configuring an Apache Web Server . . . 201

14 Managing the Database

Database Formats . . . 204
Viewing Information about the Database . . . 205
NerveCenter Database Utilities . . . 206

UNIX . . . 207
Windows . . . 208
Backing up the Database. . . . 209
Restoring the Database . . . 213
Restoring the Database Using the SerializeDB Application . . . 213
Restoring the Database from the Command Line . . . 216
Transferring Data between Databases . . . 218
ncdb2html.pl . . . 219
Troubleshooting: Managing the NerveCenter Database. . . . 224

15 Managing Management Information Bases (MIBs)

Sources of MIB Definitions . . . 226
Definitions Compiled Into the NerveCenter MIB . . . 226
Definitions Not Compiled Into the NerveCenter MIB . . . 227
Definitions From Other Sources . . . 228
Adding or Removing MIB Definitions. . . . 228
Compiling the NerveCenter MIB . . . 231
Troubleshooting: Managing MIBs . . . 233

A Error Messages

User Interface Messages. . . . 236
Error Messages . . . 238
Action Manager Error Messages. . . . 239
Alarm Filter Manager Error Messages . . . 243
Deserialize Manager Error Messages . . . 243
Flatfile Error Messages . . . 243
Inform NerveCenter Error Messages. . . . 244
Inform OV Error Messages . . . 244
LogToDatabase Manager Error Messages. . . . 246
LogToFile Manager Error Messages. . . . 247
Poll Manager Error Messages . . . 247
Protocol Manager Error Messages. . . . 247
PA Resync Manager Error Messages . . . 248
Server Manager Error Messages . . . 250
Trap Manager Error Messages. . . . 254
NerveCenter installation Error Messages (UNIX) . . . 255
OpenView Configuration Error Messages (UNIX). . . . 257

B Troubleshooting NerveCenter

Common problems 260
The NerveCenter Server 260
The NerveCenter Node List 260
Alarm Actions 261
The NerveCenter Database 261
Management Information Bases (MIBs) 261
Using Trace Counters to Troubleshoot NerveCenter 262
Using Trace Counters 263
Resetting Counters 263
Refreshing the Trace Counter Display 264
Alarm Action Counters 265
Database Counters 265
Inform Counters 266
Inform NerveCenter Counters 266
Log to Database Counters 267
Log to File Counters 267
Server Counters 268
Node Source Counters 268
Poll Counters 269
Trap Counters 270
Trigger Counters 270
Troubleshooting ASN.1 files 271
Use of Characters 271
Period, Underscore, and Hyphen 272
Capitalization 272
Table Construction 273
Duplicate Type Definitions 274
Object Identifier Format 275
Ignored IMPORT Definitions 275

C Controlling NerveCenter from the Command Line

NerveCenter Command Line Interface 278
Entering a Single Command in a UNIX Shell or a DOS Prompt 278
Entering Multiple Commands in an Interactive Session 279
Running a Script Containing One or More Commands 280
NerveCenter UNIX Shell/DOS Prompt Commands 282
client 283
dbwizard 283

importutil 284
installdb 293
ipsweep 294
mibcomp 296
mibTool 298
ncadmin 299
nccmd. 300
nclicreport 301
ncserver 301
ncstart. 303
ncstop. 304
ovpa 306
paserver 310
serializedb 313
trapgen 314
Traprcv. 318
Command Line Interface Reference 319
Server Commands 319
Object Commands 322
`add node` 322
`delete` 326
`list` 327
General Commands 329
Notes for CLI Commands 330
CLI Command Examples 330

Index. 347

Introduction

Welcome to *Managing NerveCenter*. This chapter introduces the audience and purpose of this guide, and how you can best use it.

This chapter includes the following sections:

Section	Description
Overview of this Book on page 2	Includes an overview of the contents of this guide and what you need to know before you use the guide.
NerveCenter Documentation on page 4	Lists and describes the components of the LogMatrix NerveCenter support system, including printed guides, online guides, help, and links to the LogMatrix NerveCenter Web site and the LogMatrix technical support Web site.
LogMatrix Technical Support on page 7	Describes how to access the NerveCenter knowledge base and other LogMatrix support services.

1 Introduction

Overview of this Book

Managing NerveCenter describes how NerveCenter works and how you can monitor your network most effectively. This book is written for users operating the NerveCenter Client and the NerveCenter Web Client.

Managing NerveCenter contains the following chapters:

Title	Description
Chapter 2, Understanding NerveCenter	Explains what the NerveCenter components do, how NerveCenter manages nodes, and how it can fit into a network management strategy.
Chapter 3, Administering NerveCenter	Describes the different types of NerveCenter users, their privileges, and how to administer them.
Chapter 4, Running the NerveCenter Server	Explains how to start and stop the NerveCenter server on Windows and UNIX platforms.
Chapter 5, Connecting to a NerveCenter Server	Explains how to connect a NerveCenter Administrator to a NerveCenter Server, view server status, and troubleshoot connections.
Chapter 6, Managing the NerveCenter Server	Explains how to manage NerveCenter traps and serial numbers, and how to control NerveCenter from a command line.
Chapter 7, Managing NerveCenter Port Settings	Describes how to configure NerveCenter communication, CLI, and SNMP ports.
Chapter 8, Managing Node Data	Explains how NerveCenter automates event handling; how it stores, filters, and processes node data; how to add and remove nodes; and how to troubleshoot node management.
Chapter 9, Managing SNMP Settings	Describes in detail how to configure NerveCenter for SNMP communication and support SNMPv3 agents.
Chapter 10, Managing ICMP Settings	Describes how NerveCenter uses ICMP and the controls available for configuring and monitoring this protocol.
Chapter 11, Managing NerveCenter Security	Explains how to manage security on UNIX and Windows.
Chapter 12, Managing NerveCenter Alarm Actions	Explains how to configure mail notification, dialing and paging properties, and log management settings for alarm actions.
Chapter 13, Managing NerveCenter Web Integration	Explains how to start and stop the NerveCenter Web Collector for web integration, and how to configure the web server.

Title	Description
Chapter 14, Managing the Database	Describes the NerveCenter database utilities, how to back up and restore the database, and how to transfer data between databases.
Chapter 15, Managing Management Information Bases (MIBs)	Describes the pre-compiled definitions in the NerveCenter MIB, how to add or remove definitions, and how to compile and troubleshoot the MIB.
Appendix A, Error Messages	Lists various types of NerveCenter error messages and the steps you can take to resolve them.
Appendix B, Troubleshooting NerveCenter	Describes how to troubleshoot common NerveCenter server problems.
Appendix C, Controlling NerveCenter from the Command Line	Describes how to control NerveCenter using its command line interface (CLI).

NerveCenter Documentation

This section describes the available NerveCenter documentation, which explains important concepts in depth, describes how to use NerveCenter, and provides answers to specific questions.

The documentation set is provided in online (HTML) format, as well as PDF for printing or on-screen viewing. See the following topics for more information:

- *Using the Online Help on page 4*
- *Printing the Documentation on page 4*
- *The NerveCenter Documentation Library on page 5*
- *UNIX Systems on page 6*
- *Document Conventions on page 6*
- *Documentation Feedback on page 7*

Using the Online Help

You can view the documentation with browsers such as Microsoft Internet Explorer or Firefox. Refer to the *NerveCenter Release Notes* for the browser versions supported with this release.

TIP

For in-depth instructions on using the online documentation, click the Help button ? in the upper right of the Help window.

Printing the Documentation

The NerveCenter documentation is also available as Portable Document Format (PDF) files that you can open and print. All PDF files are located in your *installpath*/doc directory.

NOTE

You must have Adobe Acrobat Reader to open or print the PDF files. You can download the Reader free from Adobe's Web Site at www.adobe.com.

The NerveCenter Documentation Library

The following documents ship with NerveCenter.

Book Title	Description	Application	Audience	PDF for Print
NerveCenter Release Notes	Describes new NerveCenter features and includes late-breaking information, software support, corrections, and instructions.	All	All	relnotes.pdf
Installing NerveCenter	Helps you plan and carry out your NerveCenter upgrades and new installations. Use the *Release Notes* in conjunction with this book.	All	Installation team	install.pdf
Managing NerveCenter	Explains how to customize and tune NerveCenter after it has been installed.	NerveCenter Administrator	Administrator	managing_ nervecenter.pdf
Integrating NerveCenter with a Network Management Platform	Explains how to integrate NerveCenter with network management platforms.	NerveCenter Administrator	Administrator	integratingNC. pdf
Learning How to Create Behavior Models	Provides step-by-step instructions and examples for creating behavior models.	NerveCenter Client	Users with administrative privileges	learningModel. pdf
Designing and Managing Behavior Models	Explains behavior models in depth, how to create or modify models, and how to manage your models.	NerveCenter Client	Users with administrative privileges	designingModels.p df
Monitoring Your Network	Explains how NerveCenter works and how you can most effectively monitor your network.	NerveCenter Client and Web Client	Users	monitoringNet. pdf
Behavior Models Cookbook	Describes each behavior model shipped with LogMatrix NerveCenter.	NerveCenter Client	Users with administrative privileges	modsCookbook. pdf
Quick reference cards	Quick reference cards provide convenient reference material for common NerveCenter tasks.	NerveCenter Client and Administrator	All	quickreference. pdf

UNIX Systems

On UNIX systems, NerveCenter man pages provide command reference and usage information that you view from the UNIX shell as with other system man pages. When you specify documentation during NerveCenter installation, the script installs nroff-tagged man pages and updates your system's MANPATH environment variable to point to the NerveCenter man page directory.

Document Conventions

This document uses the following typographical conventions:

Element	Convention	Example
Key names, button names, menu names, command names, and user entries	**Bold**	Press **Tab** Enter **ovpa -pc**
◆ A variable you substitute with a specific entry ◆ Emphasis ◆ Heading or Publication Title	*Italic*	Enter **./installdb -f** *IDBfile*
Code samples, code to enter, or application output	`Code`	`iifInOctets > 0`
Messages in application dialog boxes	Message	Are you sure you want to delete?
An arrow (>) indicates a menu selection	>	Choose **Start** > **Programs** > **OpenService NerveCenter**
A link to a section in the same book	*Blue Italic*	For more information, see *Correlating Conditions*.
A link to a section in a different book **Note:** If you are using a PDF viewer, you may need to use the **Go to Previous View** button to return to the original PDF file.	*Green Italic*	For more information, see *Correlating Conditions* in *Monitoring Your Network with NerveCenter*.

⚠ **CAUTION**

A caution warns you if a procedure or description could lead to unexpected results, even data loss, or damage to your system. If you see a caution, proceed carefully.

NOTE

A note provides additional information that might help you avoid problems, offers advice, and provides general information related to the current topic.

TIP

A tip provides extra information that supplements the current topic. Often, tips offer shortcuts or alternative methods for accomplishing a task.

If toolbar buttons are available, they are displayed in the margin next to the step in which you can use them. other shortcuts are noted as tips. Also, shortcut (accelerator) keys are displayed on application menus next to their respective options.

Documentation Feedback

LogMatrix, Inc. is committed to providing quality documentation and to helping you use our products to the best advantage. If you have any comments or suggestions, please send your documentation feedback to:

Documentation
LogMatrix, Inc.
4 Mount Royal Ave, Suite 250
Marlborough, MA 01752

documentation@logmatrix.com

LogMatrix Technical Support

LogMatrix is committed to offering the industry's best technical support to our customers and partners. You can quickly and easily obtain support for NerveCenter, our proactive IT management software.

Professional Services

LogMatrix offers professional services when customization of our software is the best solution for a customer. These services enable us, in collaboration with our partners, to focus on technology, staffing, and business processes as we address a specific need.

1 Introduction

Educational Services

LogMatrix is committed to providing ongoing education and training in the use of our products. Through a combined set of resources, we can offer quality classroom style or tailored on-site training.

Contacting the Customer Support Center

For Telephone Support

Phone: 1-800-892-3646 or 1-508-597-5300

For E-mail Support

E-mail: techsupport@logmatrix.com.

For Electronic Support

LogMatrix has a Web-based customer call tracking system where you can enter questions, log problems, track the status of logged incidents, and check the knowledge base.

When you purchased your product and/or renewed your maintenance contract, you would have received a user name and password to access the LogMatrix Call Tracking System using SalesForce. You may need to contact your contracts or NerveCenter administrator for the username and password for your account with SalesForce.

If you have not received or have forgotten your log-in credentials, please e-mail us with a contact name and company specifics at techsupport@logmatrix.com.

We are committed to providing ongoing education and training in the use of our products. Through a combined set of resources, we offer quality training to our global customer base.

For Online KnowledgeBase Access

For additional NerveCenter support information, please go the LogMatrix website www.logmatrix.com for access to the following sections of information:

- **Patches and Updates** - latest installation files, patches, and updates including documentation for NerveCenter.
- **Software Alerts** - latest software alerts relative to NerveCenter.

- **KnowledgeBase Search** - search the NerveCenter KnowledgeBase for answers to your questions whether relating to the installation, usage, or operation of NerveCenter.

For User Community Access

You can seek as well as share advice and tips with other NerveCenter users at http://community.logmatrix.com/LogMatrix/

1 Introduction

Understanding NerveCenter

2

NerveCenter™ is an application designed to monitor, manage, and automate events in complex heterogeneous networks. Through a process called event correlation, NerveCenter helps you track significant events, quickly identify the root cause of critical network problems, and initiate corrective actions.This chapter explains the core NerveCenter functionality, describes the main NerveCenter components, and explains how these they interact with each other.

This chapter contains the following sections:

Section	Description
NerveCenter Overview on page 12	Introduces NerveCenter and provides an overview of how it uses behavior models to detect important network events and track device status.
How NerveCenter Correlates Events on page 13	Introduces how NerveCenter polls and alarms play a role in identifying important network events.
How NerveCenter Manages Nodes on page 14	Introduces how NerveCenter isolates and responds to emerging network and system problems.
How Behavior Models Work on page 15	Introduces the objects that make up behavior models and describes how they interact.
NerveCenter Components on page 21	Introduces the primary NerveCenter components.
Role in Network Management Strategy on page 29	Introduces how NerveCenter can operate from a central stand-alone server, as a distributed system on multiple servers, or as a component integrated with other products.

NerveCenter Overview

There are many management tools designed to identify network faults and send alerts. However, these tools often flood the event console with events and network data. Each critical or warning message indicating a possible problem is usually accompanied by many additional messages — if a router fails, for example, the console receives many events concerning machines behind the router that cannot be contacted. Consoles also receive many messages about normal events such as simple power-up operations. As a result, there is a great deal of information to sift through before the real problem can be identified and corrected.

NerveCenter is a network management tool that helps automate the identification, tracking, management, and resolution of important network events. NerveCenter uses the Simple Network Management Protocol (SNMP) to acquire data about managed devices. NerveCenter also relies on Internet Control Message Protocol (ICMP) messages from your network to provide basic information about unresponsive devices.

At the heart of NerveCenter is the event correlation engine. NerveCenter greatly reduces the volume of event data by intelligently interpreting and correlating raw network data to determine whether a problem is serious enough to warrant human intervention. NerveCenter can then take automatic corrective action to solve the problem, inform the management platform, or notify the appropriate person.

For each device that NerveCenter monitors, it creates one or more finite state machines—or alarms—that define operational states that it wants to detect, and defines rules that affect transitions between operational states. These rules can be very simple; for example, a state transition can be caused by the receipt of a generic Simple Network Management Protocol (SNMP) trap. Or they can be quite complex and take advantage of NerveCenter's support for Perl expressions.

In addition to being an advanced event automation solution, NerveCenter is a highly scalable, cross-platform client/server application. It can run co-resident with a network management platform (such as Hewlett Packard's OpenView Network Node Manager) and manage thousands of nodes, or the server can run as a background process at tens or even hundreds of remote offices.

How NerveCenter Correlates Events

NerveCenter obtains data from SNMP agents running on managed nodes by processing incoming SNMP traps and polling the nodes for specific MIB values using SNMP Get requests. When a predefined network condition is detected, NerveCenter stores the event information in a finite state machine called an alarm. The alarm continues to track the status of the interface, node, or enterprise being monitored. The alarm waits for subsequent events or issues polls to determine if the condition warrants further action.

For example, by tracking events based on their persistence, NerveCenter can detect when a router experiences a persistent link-down condition. Upon detecting a downed communication link, NerveCenter waits to see if the link is restored within a given amount of time. If the link comes back up within the specified time, NerveCenter acknowledges the situation and returns the alarm to its normal (ground) state. However, if the link remains down, NerveCenter can perform a variety of actions based on your management strategy. Also, NerveCenter can detect when a group of nodes appears to be down or unreachable and then poll their parent router. If the router is down, further polling of those nodes is disabled until the router is back up.

To correlate and filter this data, NerveCenter relies on configurable network and system behavior models for each type of managed resource. A behavior model is a group of NerveCenter objects that detect and handle a particular network or system behavior. A typical behavior model consists of an alarm with its supporting polls and masks, though behavior models can have multiple alarms. Any managed device can be associated with one or more behavior models. For a description of behavior models, see *How Behavior Models Work on page 15*.

Once a NerveCenter behavior model has identified a problem, it can take automatic actions, including notifying an administrator or a network management platform, executing a program or script, modifying the node’s properties, changing SNMP values, and logging the critical data.

How NerveCenter Manages Nodes

To automate events, NerveCenter relies on the definition of *behavior models*. These models are constructed from NerveCenter objects and define:

- Which devices, or nodes, a behavior model will affect
- How NerveCenter will detect certain conditions on these nodes
- How NerveCenter will correlate the conditions it detects
- How NerveCenter will respond to network problems

NerveCenter can get the list of nodes to monitor from a network management platform, discover them on the network, or import the information from another NerveCenter database. NerveCenter most often obtains data from SNMP (Simple Network Management Protocol) agents running on managed nodes.

NerveCenter assigns a set of properties to each managed node that determine the behavior models that can be applied. Properties typically describe the type of the device—for example, a router—or are named after objects in the management information base (MIB) used to manage the node.

After gathering data, the NerveCenter correlation engine examines detected network conditions and determines how the conditions may be related, or the underlying cause that produced the conditions. Detected conditions can be correlated in many ways, which include detecting the persistence of a condition, finding a set of conditions, and looking for a sequence of conditions. For instance, NerveCenter may scan a large number of events and identify a subset that relate to SNMP authentication failures on a managed node. NerveCenter may then determine that the authentication failures were far enough apart that no problem exists, or it may find that several failures occurred within a short period of time — which may indicate a possible security problem.

NerveCenter not only enables you to detect network and system problems, but is able to respond automatically to the conditions it detects. To set up these automated responses, you associate *actions* with state transitions. For example, NerveCenter can notify an administrator via email or pager, log the information to a file, trigger a transition in another alarm, or execute a script.

For more information on nodes, see *Chapter 8, Managing Node Data*.

How Behavior Models Work

NerveCenter detects events and compares these events to conditions of interest defined in behavior models. To understand how behavior models work, you must understand the objects that make up behavior models and how they interact.

Behavior Model Objects

NerveCenter uses the objects listed in *Table 2-1* to define behavior models:

TABLE 2-1. NerveCenter Objects Used with Behavior Models

Object	Description
Node	Represents either a workstation, server, or a network device such as a router, hub, or bridge.
Property	A text string that describes the type of node or one of the node's MIB objects. Polls and alarms use properties to target specific nodes.
Property group	A group of properties assigned to a node. The properties in the group allow behavior models to target the node.
Trigger	An internal event that is generated when a defined network event has been detected or a condition satisfied. Triggers are generated by polls and trap masks, as well as by other NerveCenter objects such as Perl subroutines and NerveCenter alarms.
Poll	Periodically solicits defined SNMP values from the agents running on targeted nodes and fires a trigger when specified conditions are met. Polls can fire multiple triggers, which can be detected by one or more alarms.
Trap mask	Detects a predefined type of SNMP trap. Trap masks can specify generic and specific trap numbers, enterprise OIDs, and variable binding data to match against incoming traps. A trap mask fires a trigger when the specified type of trap is detected.
Alarm	Detects triggers that cause the alarm to transition from one state to the next. Each transition is triggered by its own set of network data, which is defined in the poll or mask that generates the associated trigger. The alarm performs any actions assigned to a transition.

Behavior Model Operation

A behavior model is not a NerveCenter object. Rather, it is a collection of NerveCenter objects designed to monitor specific network devices, events, or behaviors. *Figure 2-1* shows the interaction of objects in a simple behavior model.

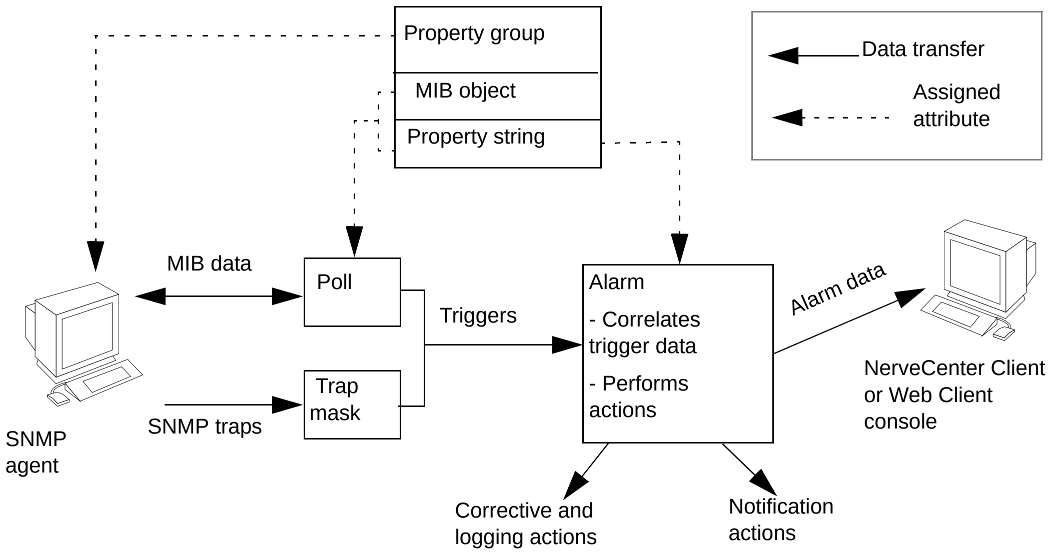


FIGURE 2-1. A Simple Behavior Model

When a trigger-generating object detects an event of interest on a node, it raises a trigger. This trigger generator might be a NerveCenter poll that looks for a specified network condition, or a trap mask that detects a certain type of SNMP trap. NerveCenter alarms internally subscribe for triggers based on alarm states. When an alarm detects its first trigger, the alarm transitions to a predefined state. Based on the state, the alarm ignores triggers that can no longer cause transitions and listens instead for those triggers that can. The alarm waits in that state until another trigger is received—either from the same or another trigger generator. This allows an alarm to react only to relevant events and poll only for relevant data. For more information about alarms, see *Alarms on page 18*.

How Properties Affect Behavior Models

Properties allow you to limit excessive polling and control which nodes are targeted by which alarms. A property is text string that describes the type of node or one of the node's MIB objects. A property group is a group of properties assigned to a node. Each node is targeted based on its assigned property group, which can contain many properties. Before an alarm can be applied to a node, the node's group of properties must include any property specified in the alarm or its associated polls. Additionally, before a node can be polled, the node's property group must contain the MIB base object that the poll is designed to evaluate. Assigning a node to a property group with multiple properties allows the node to be targeted by multiple alarms.

Other Trigger Generators

Alarm transitions can trigger actions that, themselves, generate triggers. Trigger-generating actions can cause a transition in another alarm or transition yet another state in the same alarm. An alarm can also execute a Perl subroutine, which can fire a trigger when certain conditions are met. In both cases, the alarm serves as a trigger generator that serves an important function in a behavior model. These types of trigger generators provide an additional level of control over your alarms.

Figure 2-2 shows the interaction of various trigger generators in a behavior model.

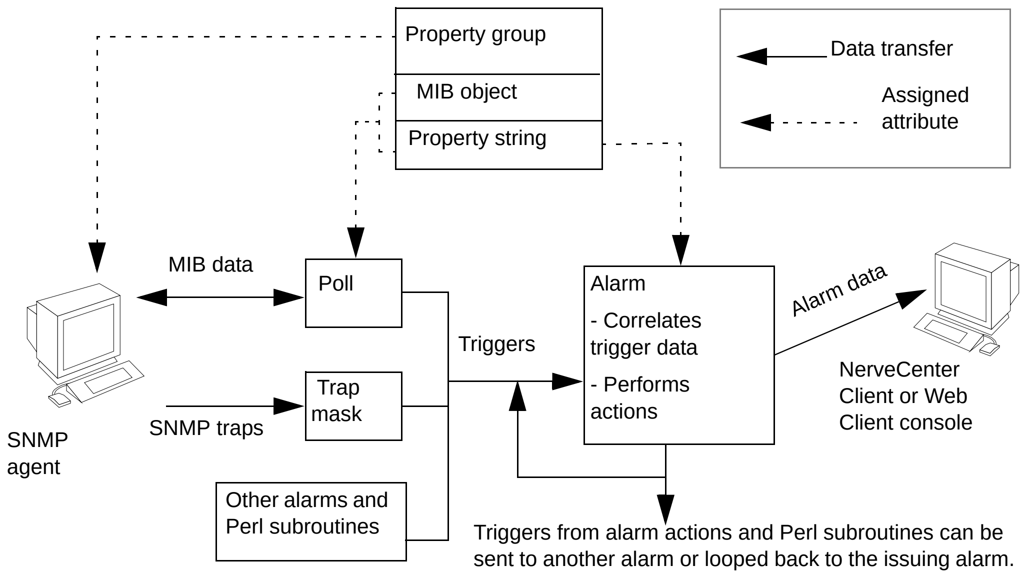

Property group
MIB object
Property string
Data transfer
Assigned attribute
MIB data
Poll
Alarm
- Correlates trigger data
- Performs actions
Alarm data
Triggers
Trap mask
SNMP traps
SNMP agent
NerveCenter Client or Web Client console
Other alarms and Perl subroutines
Triggers from alarm actions and Perl subroutines can be sent to another alarm or looped back to the issuing alarm.

FIGURE 2-2. A Behavior Model with Triggers Generated by Alarms

Alarm Actions

NerveCenter can implement a broad range of actions when an alarm is instantiated. A transition might cause NerveCenter to send a trap, email, or page. NerveCenter has its own type of message—an inform—that contains the variable bindings associated with the event that caused the alarm to transition. Informs can be sent to one or more destinations, including network management platforms and other NerveCenters. For each destination, you can specify a minimum severity level that your network conditions must meet before the informs are issued.

NerveCenter can call a Perl subroutine as an alarm action. The Perl subroutine might fire a trigger when certain conditions are met or execute other commands that effect changes to your network. Perl subroutines have access to associated alarm and trigger data for nodes that cause alarms to transition. Subroutines can then evaluate this information and conditionally take further action.

NerveCenter can also log alarm data and perform corrective actions such as setting SNMP attribute values. Finally, all actions can be performed conditionally using the NerveCenter Action Router. The Action Router performs actions based on user-defined criteria such as time of day, severity of the alarm, type of node, and so on.

Alarms

Alarms are key to behavior model operation. An alarm is a finite state machine that defines the operational states it wants to detect and the transitions from one state to the next when the proper trigger is received. Each transition is triggered by its own set of network data, which is defined in the NerveCenter object that generates the associated trigger. Transitions can also be driven by any alarm action that causes a trigger to be fired. If actions are associated with a transition, the server performs these actions each time the transition takes place.

Figure 2-3 shows the state diagram for an alarm called IfUpDownStatus. IfUpDownStatus monitors the operational status of interfaces on managed nodes. If an interface is down, an inform action notifies the network management platform.

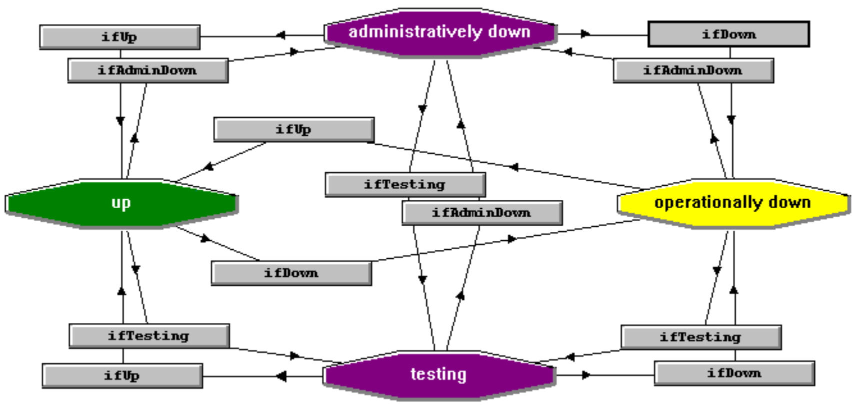


FIGURE 2-3. IfUpDownStatus Alarm

Each state is depicted as an icon in the diagram. The state icons are color coded to represent a particular level of severity.

The alarm contains four states related to the status of an interface:

- **Up**—The interface is up. The current (operational) and desired (administrative) statuses are set to 1 (defined in RFC1213 as up). This state has a severity level of Normal.
- **Administratively down**—Both the current (operational) and the desired (administrative) statuses are set to 2, defined in RFC1213 as down.
- **Testing**—Either the current (operational) or desired (administrative) status is set to 3, defined in RFC1213 as test mode.
- **Operationally down**—The current (operational) status is down but the desired (administrative) status is up. This state has a high severity level of Major. All transitions leading to this state include an alarm action that sends an inform to NerveCenter or to the platform.

The ifStatus poll fires all the triggers that transition the IfUpDownStatus alarm. The poll is designed to evaluate the MIB base object attributes ifAdminStatus and ifOperStatus, which belong to the ifEntry base object in the interfaces MIB-II group. The poll fires four triggers—one for each of the four states previously described—based on the values returned for the two attributes. The alarm correlates the data it receives from the poll and transitions based on the combinations shown in *Table 2-2*.

TABLE 2-2. Correlation of IfEntry Attributes in the IfUpDownStatus Alarm

IfUpDownStatus Status	ifAdminStatus	ifOperStatus
Up	1	1
Administratively down	2	2
Testing	3	3
Operationally down	1	2

In this alarm, a trigger representing a condition other than up for either attribute—ifAdminStatus or ifOperStatus—transitions the alarm from the Up state. A subsequent trigger representing a condition other than up for the remaining attribute then transitions the alarm to a different state. The state to which it transitions depends on the combination of the values for the two attributes.

To learn more about NerveCenter's behavior models, refer to the following sources:

- *Learning How to Create Behavior Models* and *Designing and Managing Behavior Models* contain complete descriptions and procedures for behavior models.
- NerveCenter objects, such as alarms, include notes that describe the object and provide specific information related to that object. For example, the notes for an alarm include a list of triggers that transition the alarm, the objects that fire those triggers, and any actions associated with the alarm's transitions.

NerveCenter Components

This section introduces the main NerveCenter components:

- *NerveCenter Server on page 21*
- *NerveCenter Database on page 21*
- *NerveCenter MIB on page 22*
- *NerveCenter Interfaces on page 23*

NerveCenter Server

The NerveCenter Server is responsible for carrying out all of the major tasks that NerveCenter performs. For example, it handles the polling of SNMP agents, creates NerveCenter objects such as the finite alarms mentioned earlier, and makes sure that state transitions occur at the appropriate times. The server also performs all actions associated with state transitions.

The server can run as a daemon on UNIX systems and as a service on Windows systems. This capability to run in the background has important implications with regard to using NerveCenter at remote sites. You can install the server and database at a remote office and have that server manage the local network, yet control the server (via the NerveCenter Client) from a central location. Servers located at remote sites can forward noteworthy information to a server at the central location as required.

NerveCenter Database

The NerveCenter database is primarily a repository for the NerveCenter objects that make up a set of behavior models; the basic objects are introduced in *Behavior Model Objects on page 15*. A set of objects that define many useful behavior models ships with NerveCenter and is available as soon as you've installed the product. On UNIX systems, the NerveCenter database is implemented as a flat file. On Windows systems, the database can be either a Microsoft Access database or a Microsoft SQL Server database.

NerveCenter MIB

NerveCenter enables you to control which types of devices are managed based on their SNMP management information base (MIB) definitions. *Figure 2-4* shows the NerveCenter MIB files along with the other NerveCenter components.

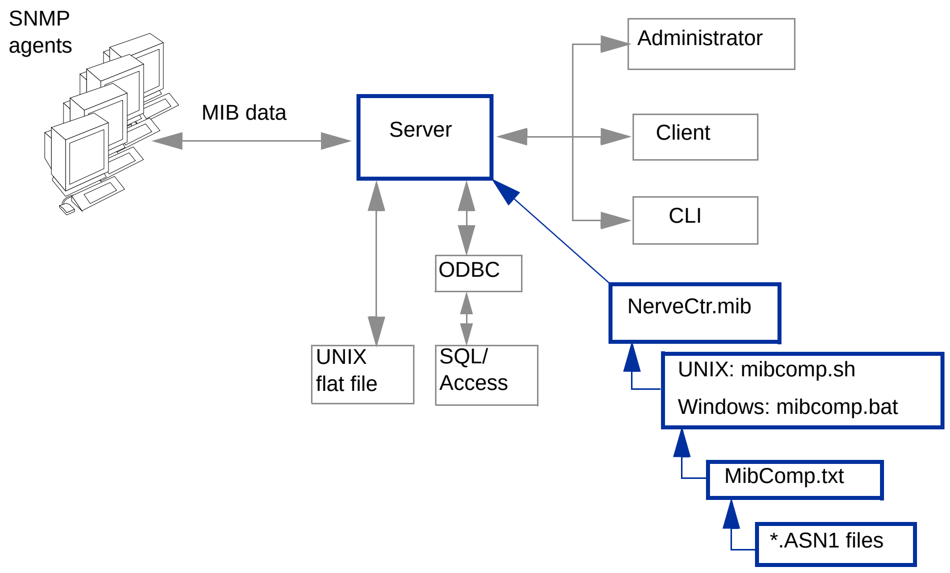


FIGURE 2-4. NerveCenter MIB Files

The compiled MIB file shipped with NerveCenter, NerveCtr.mib, contains definitions for Internet Standard RFC SNMP Versions 1, 2 and 3 agents as well as many vendor specific MIBs. NerveCtr.mib defines the management information available from an SNMP agent based on the MIBs the agent supports. This support information is contained in the .ASN1 file for each managed device. NerveCenter's MIB file enables you to construct SNMP requests for particular devices and to decode data received in SNMP traps.

NerveCtr.mib is compiled from definitions referenced in a text file, which by default is named MibComp.txt. MibComp.txt, in turn, points to the *.ASN1 files you are using. To add support for new devices, modify MibComp.txt to include the new *.ASN1 files, located within the *installation*/MIB directory. You must have the proper .ASN1 file for each device that is defined.

After modifying the text file, you recompile the MIB file. NerveCenter's MIB compiler, mibcomp, verifies each respective .ASN1 file, compiles the definitions in the text file, and creates a new version of NerveCtr.mib that can then be loaded into the NerveCenter Server.

NerveCenter Interfaces

This section introduces the four principal user interfaces to the NerveCenter server:

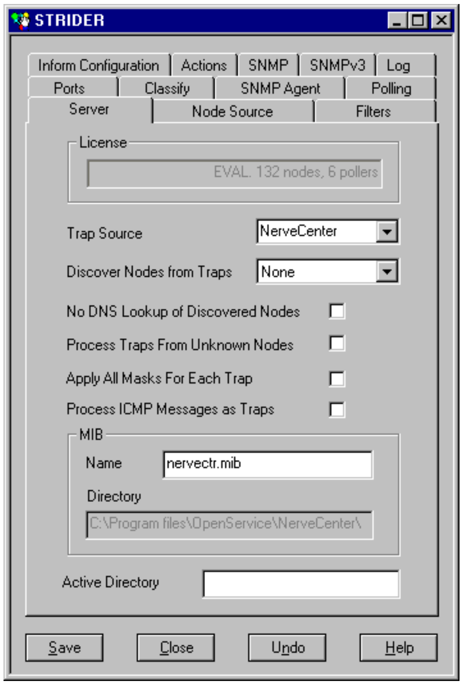
- *The NerveCenter Administrator on page 24*
- *The NerveCenter Client on page 26*
- *The NerveCenter Web Client on page 27*
- *The Command Line Interface on page 28*

The NerveCenter Administrator

Users with administrator privileges can use the NerveCenter Administrator interface to:

- Configure NerveCenter discovery mechanisms
- Configure the number of SNMP polling retries and the retry interval
- Configure NerveCenter mail and paging actions
- Manage NerveCenter log files
- Configure NerveCenter to work with a network management platform

Figure 2-5 shows the NerveCenter Administrator.



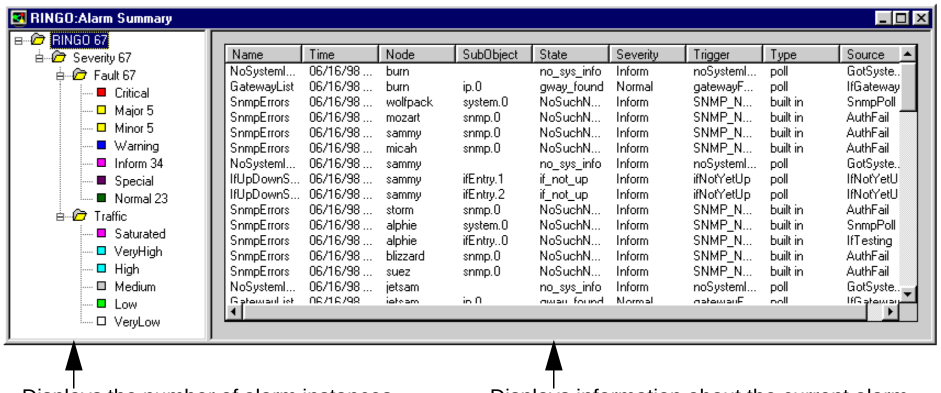
License: EVAL. 132 nodes, 6 pollers
Trap Source: NerveCenter
Discover Nodes from Traps: None
No DNS Lookup of Discovered Nodes ☐
Process Traps From Unknown Nodes ☐
Apply All Masks For Each Trap ☐
Process ICMP Messages as Traps ☐
MIB
Name: nervectr.mib
Directory: C:\Program files\OpenService\NerveCenter\
Active Directory:

FIGURE 2-5. NerveCenter Administrator

The NerveCenter Client Console

The NerveCenter Client lets you monitor the network as well as create and modify the behavior models managed by the NerveCenter Server. You can access the MIBs, nodes, behavior models, alarm filters, and other NerveCenter objects associated with that Server. The Alarm Summary window displays all current alarms for the connected Server.

Figure 2-6 shows a sample Alarm Summary window.



Name	Time	Node	SubObject	State	Severity	Trigger	Type	Source
NoSystemI...	06/16/98 ...	burn		no_sys_info	Inform	noSystemI...	poll	GotSyste..
GatewayList	06/16/98 ...	burn	ip.0	gway_found	Normal	gatewayF...	poll	IfGateway
SnmpErrors	06/16/98 ...	wolfpack	system.0	NoSuchN...	Inform	SNMP_N...	built in	SnmpPoll
SnmpErrors	06/16/98 ...	mozart	snmp.0	NoSuchN...	Inform	SNMP_N...	built in	AuthFail
SnmpErrors	06/16/98 ...	sammy	snmp.0	NoSuchN...	Inform	SNMP_N...	built in	AuthFail
SnmpErrors	06/16/98 ...	micah	snmp.0	NoSuchN...	Inform	SNMP_N...	built in	AuthFail
NoSystemI...	06/16/98 ...	sammy		no_sys_info	Inform	noSystemI...	poll	GotSyste..
IfUpDownS...	06/16/98 ...	sammy	ifEntry.1	if_not_up	Inform	ifNotYetUp	poll	IfNotYetU
IfUpDownS...	06/16/98 ...	sammy	ifEntry.2	if_not_up	Inform	ifNotYetUp	poll	IfNotYetU
SnmpErrors	06/16/98 ...	storm	snmp.0	NoSuchN...	Inform	SNMP_N...	built in	AuthFail
SnmpErrors	06/16/98 ...	alphie	system.0	NoSuchN...	Inform	SNMP_N...	built in	SnmpPoll
SnmpErrors	06/16/98 ...	alphie	ifEntry..0	NoSuchN...	Inform	SNMP_N...	built in	IfTesting
SnmpErrors	06/16/98 ...	blizzard	snmp.0	NoSuchN...	Inform	SNMP_N...	built in	AuthFail
SnmpErrors	06/16/98 ...	suez	snmp.0	NoSuchN...	Inform	SNMP_N...	built in	AuthFail
NoSystemI...	06/16/98 ...	jetsam		no_sys_info	Inform	noSystemI...	poll	GotSyste..

Displays the number of alarm instances for each severity level.

Displays information about the current alarm instances for the folder selected in the left pane.

FIGURE 2-6. Alarm Summary Window

The left pane contains a tree that displays the total number of current alarm instances, the number of instances in each severity group (Fault and Traffic), and the number of instances of each severity. If there is no number next to a severity, there are no active alarm instances of that severity.

The right pane provides detailed information about current alarm instances for the folder or severity that is highlighted in the tree view.

The NerveCenter Client

Figure 2-7 shows the GUI for the NerveCenter Client.

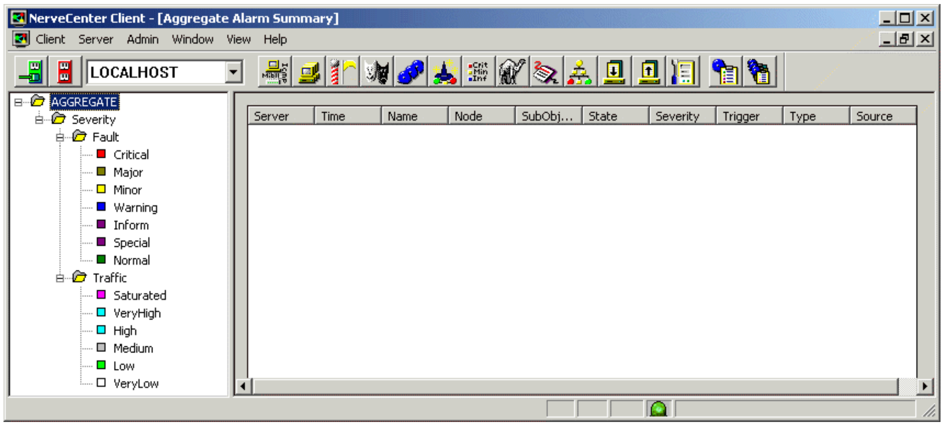


FIGURE 2-7. NerveCenter Client

Two types of users run the NerveCenter Client. Users with NerveCenter User privileges can run the client to:

- Monitor active alarms
- Filter alarms for the alarm summary windows
- View an alarm's history
- Reset alarms
- Monitor the state of managed nodes
- Generate reports

For complete information on using the NerveCenter Client to perform the tasks listed above and others, see *Monitoring Your Network*.

Users with NerveCenter Administrator privileges can perform all the tasks that users with User privileges can. In addition, they can use the client to:

- Create new behavior models
- Customize the predefined behavior models
- Modify, copy, or delete any object in the NerveCenter database

The NerveCenter Web Client

NerveCenter provides a Web-based console, called the NerveCenter Web Client, that lets you monitor network activity from any machine that has a Microsoft Internet Explorer or Netscape Navigator browser installed. In order to use a Web-based console, you must install NerveCenter's Web support component, the Web Collector. The Web Collector can be installed on the NerveCenter Server host or on your Web server.

Figure 2-8 shows how the Web Collector works with NerveCenter components.

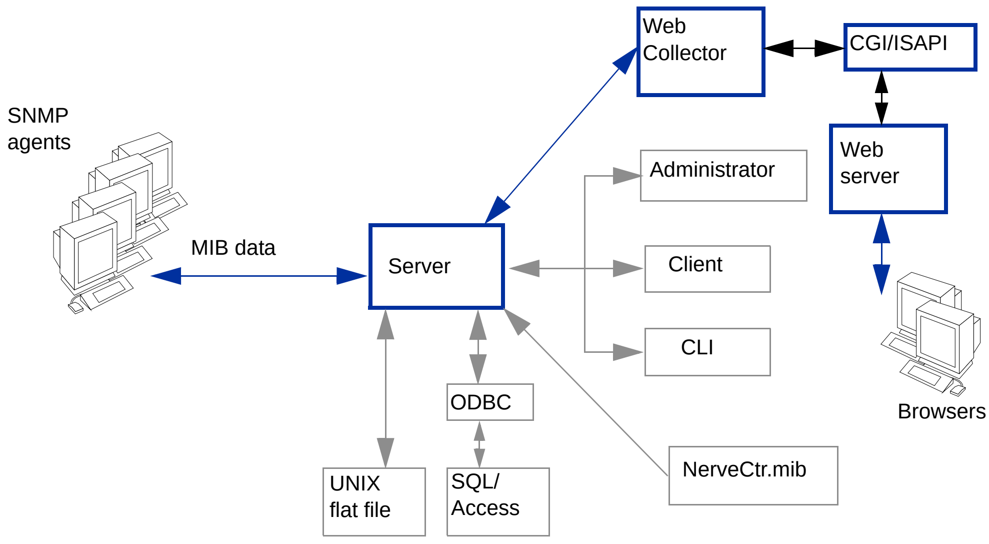
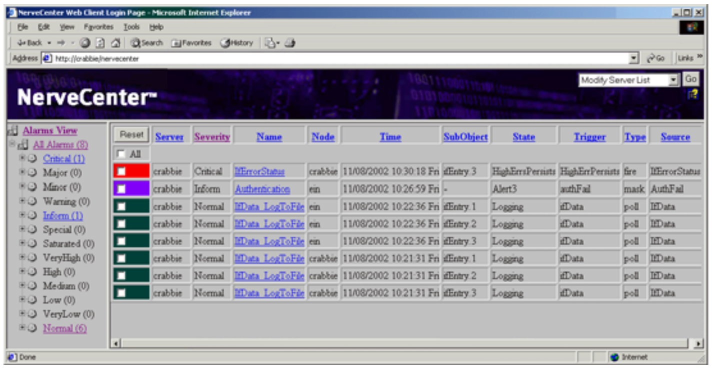


FIGURE 2-8. Web Server Integration

You can monitor NerveCenter alarms from a browser by entering http://*webserver*/NerveCenter, where *webserver* is the name of the machine on which your Web server is running. After entering this URL, your Web server invokes a NerveCenter CGI/ISAPI application that connects you to the Web Collector. The Web Collector is responsible for transmitting information between the NerveCenter Server and the Web Client browser. Through the Web Collector, the NerveCenter Server prompts you to enter your NerveCenter user name and password. The NerveCenter Server verifies your logon and, in turn, provides alarm data. This alarm information is relayed through the Web Collector, CGI application, and Web server to your browser.

Figure 2-9 shows the GUI for the NerveCenter Web Client.



	Server	Severity	Name	Node	Time	SubObject	State	Trigger	Type	Source
🟥	crabbie	Critical	IfErrorStatus	crabbie	11/08/2002 10:30:18 Fri	ifEntry 3	HighErrsPersists	HighErrPersists	fire	IfErrorStatus
🟪	crabbie	Inform	Authentication	ein	11/08/2002 10:26:59 Fri	-	Alert3	authFail	mark	AuthFail
🟩	crabbie	Normal	IfData_LogToFile	ein	11/08/2002 10:22:36 Fri	ifEntry 1	Logging	ifData	poll	IfData
🟩	crabbie	Normal	IfData_LogToFile	ein	11/08/2002 10:22:36 Fri	ifEntry 2	Logging	ifData	poll	IfData
🟩	crabbie	Normal	IfData_LogToFile	ein	11/08/2002 10:22:36 Fri	ifEntry 3	Logging	ifData	poll	IfData
🟩	crabbie	Normal	IfData_LogToFile	crabbie	11/08/2002 10:21:31 Fri	ifEntry 1	Logging	ifData	poll	IfData
🟩	crabbie	Normal	IfData_LogToFile	crabbie	11/08/2002 10:21:31 Fri	ifEntry 2	Logging	ifData	poll	IfData
🟩	crabbie	Normal	IfData_LogToFile	crabbie	11/08/2002 10:21:31 Fri	ifEntry 3	Logging	ifData	poll	IfData

FIGURE 2-9. NerveCenter Web Client

The NerveCenter Web Client, unlike the NerveCenter Client, is meant to be used only for monitoring a network, not for creating behavior models. It enables you to:

- Monitor active alarms
- View an alarm's history
- Reset alarms
- Monitor the state of managed nodes

For complete information on using the web client, see *Monitoring Your Network*.

The Command Line Interface

You can use NerveCenter's command line interface (CLI) to delete, list, or set (enable or disable) alarms, trap masks, nodes, and polls from a Windows Command Prompt or a UNIX shell. You can also connect to, display the status of, and disconnect from NerveCenter servers using the CLI. You can issue commands manually or from a script.

See *Appendix C, Controlling NerveCenter from the Command Line* for command descriptions.

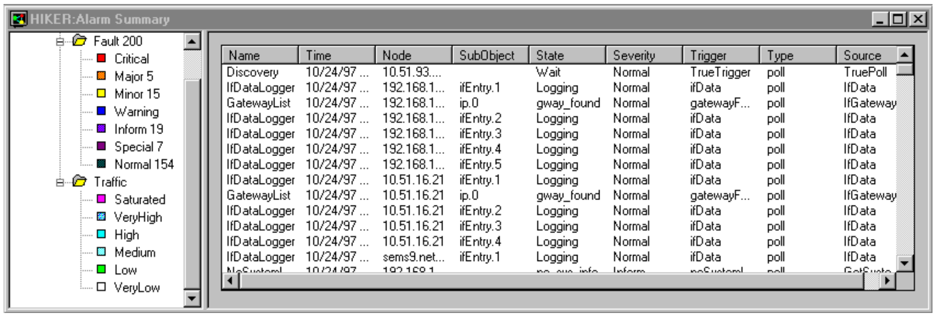
Role in Network Management Strategy

NerveCenter can play a variety of roles in an overall network management strategy. The role that NerveCenter plays in your strategy depends largely on the size of your network and on what other products you are using to manage your network and systems:

- If you are managing a small network, NerveCenter can be used as a standalone system. It can discover the workstations and network devices on the network, detect and correlate network conditions, respond automatically to conditions, and display information about active alarms. See the section *Standalone Operation on page 30* for further information.
- For larger networks, multiple NerveCenters can be used in concert. Local NerveCenter systems could be set up to manage remote sites, and the local NerveCenter servers could forward important information to the NerveCenter server at the central site. See the section *Using Multiple NerveCenter Servers on page 31* for further information.
- NerveCenter can be used in conjunction with a network management platform such as Hewlett Packard OpenView Network Node Manager, and IBM Tivoli Netcool/OMNIbus which manage systems, networks, intranets, and databases. NerveCenter can be configured to receive messages from or send messages to these network management platforms. See the section *Integration with Network Management Platforms on page 32* for further information.

Standalone Operation

At smaller sites, you can use NerveCenter alone for your network management tasks. As we’ve seen, NerveCenter is very strong in the areas of event correlation and automated actions. In addition, NerveCenter includes an alarm console, as shown in *Figure 2-10*.



Name	Time	Node	SubObject	State	Severity	Trigger	Type	Source
Discovery	10/24/97 ...	10.51.93....		Wait	Normal	TrueTrigger	poll	TruePoll
IfDataLogger	10/24/97 ...	192.168.1...	ifEntry.1	Logging	Normal	ifData	poll	IfData
GatewayList	10/24/97 ...	192.168.1...	ip.0	gway_found	Normal	gatewayF...	poll	IfGateway
IfDataLogger	10/24/97 ...	192.168.1...	ifEntry.2	Logging	Normal	ifData	poll	IfData
IfDataLogger	10/24/97 ...	192.168.1...	ifEntry.3	Logging	Normal	ifData	poll	IfData
IfDataLogger	10/24/97 ...	192.168.1...	ifEntry.4	Logging	Normal	ifData	poll	IfData
IfDataLogger	10/24/97 ...	192.168.1...	ifEntry.5	Logging	Normal	ifData	poll	IfData
IfDataLogger	10/24/97 ...	10.51.16.21	ifEntry.1	Logging	Normal	ifData	poll	IfData
GatewayList	10/24/97 ...	10.51.16.21	ip.0	gway_found	Normal	gatewayF...	poll	IfGateway
IfDataLogger	10/24/97 ...	10.51.16.21	ifEntry.2	Logging	Normal	ifData	poll	IfData
IfDataLogger	10/24/97 ...	10.51.16.21	ifEntry.3	Logging	Normal	ifData	poll	IfData
IfDataLogger	10/24/97 ...	10.51.16.21	ifEntry.4	Logging	Normal	ifData	poll	IfData
IfDataLogger	10/24/97 ...	sems9.net...	ifEntry.1	Logging	Normal	ifData	poll	IfData

FIGURE 2-10. NerveCenter’s Alarm Console

This console displays information about every current alarm instance. In addition, if you double-click on a line in the event console, you are taken to an Alarm History window that displays information about all of the alarm transitions that have occurred for the alarm instance you selected.

At small installations, no discovery mechanism is necessary; you can add nodes to NerveCenter manually. At somewhat larger sites, however, such a mechanism is helpful, and NerveCenter provides one in its Discovery behavior model.

Using Multiple NerveCenter Servers

Because a NerveCenter server can inform another NerveCenter server or management platform of a network condition, it’s possible to set up NerveCenter servers at remote sites that notify a centrally located NerveCenter server or management platform (*Figure 2-11*).

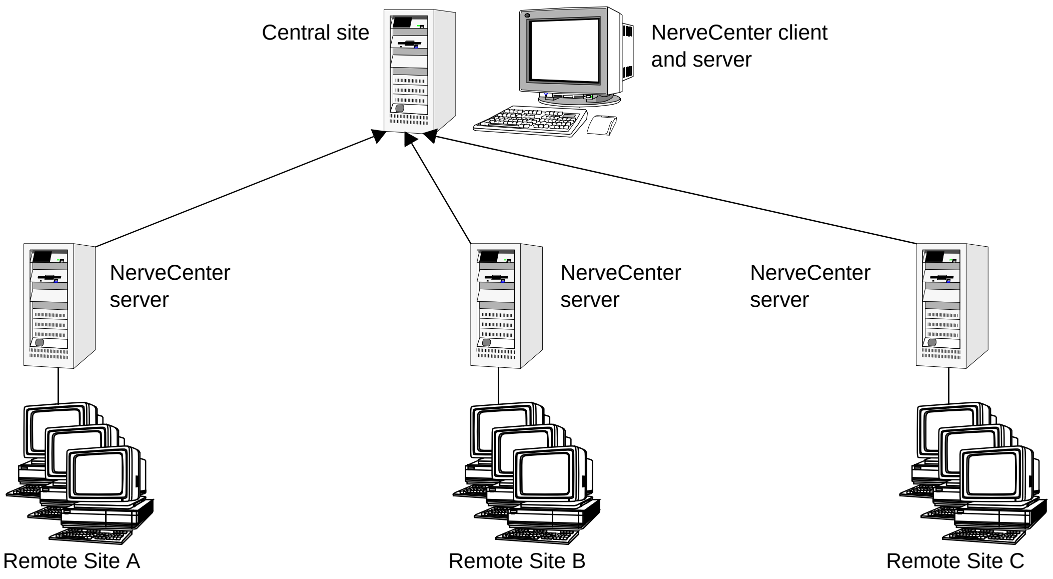


FIGURE 2-11. Distributed NerveCenter Servers

This is a reliable solution because the remote NerveCenter servers use TCP/IP to notify the central NerveCenter server and retransmit messages as necessary to ensure their delivery. There are a few advantages to this type of setup:

- Only a small amount of data is transmitted over the WAN. Any bandwidth intensive monitoring is conducted on a LAN and is managed by a remote NerveCenter server.
- Remote NerveCenter servers can be run in lights-out mode, which means that:
  - NerveCenter runs as a Windows service or as a UNIX daemon
  - You can monitor and configure NerveCenter from a remote location
  - You can modify all NerveCenter parameters without shutting NerveCenter down
  - No display or operators are required at a site
- The central NerveCenter can further correlate and filter conditions across remote NerveCenter Server domains

Integration with Network Management Platforms

A network management platform (NMP) is an operations and problem-management solution for use in a distributed multi-vendor environment. Intelligent distributed agents on managed nodes monitor system and application log files and SNMP data. The agents apply filters and thresholds to monitored data and forward messages about conditions of interest to a central management station. When the management station receives messages, it can automatically take corrective action—such as broadcasting a command to a set of systems—or an operator can initiate a response.

Additionally, with OpenView Network Node Manager, you can direct NerveCenter to take its node information from the management platform and configure NerveCenter to take over all polling activity and event processing.

You can integrate your NerveCenter installation with the NMP so that the NMP can send messages to NerveCenter for correlation or processing. After the messages arrive, NerveCenter correlates the conditions described in these messages with related conditions—from the NMP or from other sources—and can respond with any of its alarm actions, as appropriate. In addition, NerveCenter can send a message to an NMP in response to any network condition, whether the condition was originally detected by the NMP or not.

NMPs alone can detect a condition and invoke an action in response. However, you must integrate the NMP with NerveCenter if you want to:

- Correlate conditions detected by the NMP on different devices
- Correlate different types of conditions detected by the NMP on the same device
- Correlate conditions detected by the NMP with other types of events or conditions on the same device or across different devices

See *Integrating NerveCenter with a Network Management Platform* for more information.

Administering NerveCenter

NerveCenter is designed to run in the background, quietly monitoring your network. Fortunately, it does not require a great deal of administrative attention. There are, however, some initial settings as well as ongoing maintenance issues that must be attended to. A person designated as a NerveCenter administrator usually handles these tasks.

This chapter includes the following sections:

Section	Description
Who Uses NerveCenter? on page 34	Explains the difference between basic users, power users, and administrators of NerveCenter.
NerveCenter Login Rights on page 35	Explains the difference between login rights as a user and as an administrator.
The Role of a NerveCenter Administrator on page 36	Recommends habits a NerveCenter administrator should develop.

Who Uses NerveCenter?

Although not necessary, you may find it helpful to think of three different groups who use NerveCenter:

- **basic users** – monitor NerveCenter using either NerveCenter Client or the Web Client. They have user login rights and should read *Monitoring Your Network*.
- **power users** – design and manage behavior models using the NerveCenter Client. They have administrator login rights and should read *Designing and Managing Behavior Models* and *Learning How to Create Behavior Models*.
- **administrators** – manage the administrative tasks related to NerveCenter. They have administrator login rights and use the NerveCenter Administrator tool. A NerveCenter administrator should read this book.

Table 3-1 illustrates the differences between the three.

TABLE 3-1. The Differences between Three Groups of NerveCenter Users

	Primary Task	Primary Tool	Rights*	Book
Basic user	Monitoring NerveCenter	NerveCenter Client; Web Client	User	*Monitoring Your Network*
Power user	Designing and managing behavior models	NerveCenter Client	Administrator	*Designing and Managing Behavior Models* *Learning How to Create Behavior Models*
Administrator	Administering NerveCenter	NerveCenter Administrator	Administrator	*Managing NerveCenter*

NerveCenter Login Rights

By default, during installation NerveCenter establishes two groups with different login rights:

- NerveCenter users
- NerveCenter administrators

Only those with NerveCenter administrators login rights can start the NerveCenter Administrator application.

Those with either NerveCenter administrators and users login rights can start the NerveCenter Client application. However, opening under NerveCenter Users login rights limits what you can do with the Client.

Table 3-2 illustrates the difference between NerveCenter administrators and NerveCenter users login rights.

TABLE 3-2. User and Administrator Login Rights in NerveCenter Client

	User	Administrator
Monitor active alarms	✓	✓
View an alarm's history	✓	✓
Reset alarms	✓	✓
Monitor the state of managed nodes	✓	✓
Generate reports	✓	✓
Create new behavior models		✓
Customize the predefined behavior models		✓
Add login rights to another NerveCenter user		✓
Modify, copy, or delete an object in the NerveCenter database		✓

The Role of a NerveCenter Administrator

Because NerveCenter is intended to run in the background, quietly monitoring your network, it does not require a great deal of administrative maintenance. There are, however, a few regular habits a NerveCenter administrator should develop. These include:

- Backing up the NerveCenter database. See *Backing up the Database on page 209*.
- Checking to see that there is available disk space for each log file.
- Checking log files to make sure they are truncating to the correct size. See *Specifying Settings for Log Management on page 191*.
- Checking for logged errors or messages in the event viewer for Windows or the system log files for UNIX. These errors or messages could indicate problems that could be currently affecting NerveCenter performance or could later develop into a problem.
- Checking all your NerveCenter Server's Server Status pages to make sure all necessary connections are available. See *Viewing NerveCenter Server Status on page 46*. You will want to monitor information on the following tabs:
  - Node Source: Is the connection with the node data source still available?
  - Inform Configuration: Is the inform recipient list current?
  - Connected NerveCenter: Does this list show all the NerveCenter Servers that may potentially send the active server an inform packet?
- Updating any relevant changes in IP addresses, such as the location of a node data source or inform recipients.
- Keeping various NerveCenter login rights current to reflect changes in personnel. See *Managing NerveCenter Security on page 165*.
- Maintaining your network management platform with good procedures. For example, in Network Node Manager running **ovtopofix -a** weekly to reconcile the Network Node Manager databases.
- Monitor the LogMatrix Software Technical Support Web for news ideas and procedures at www.logmatrix.com.

Running the NerveCenter Server

4

Before a client can connect to a NerveCenter Server, the Server must be installed and running. See *Installing NerveCenter* for instructions on installing the NerveCenter Server and other applications. This chapter explains how to start and stop the NerveCenter Server.

This chapter includes the following sections:

Section	Description
Running the NerveCenter Server on UNIX on page 38	Explains how to start and stop the NerveCenter Server on a UNIX platform.
Running the NerveCenter Server on Windows on page 39	Explains how to start and stop the NerveCenter Server on a Windows platform.
Troubleshooting: Running the NerveCenter Server on page 40	Lists common problems users face when running the NerveCenter Server.

Running the NerveCenter Server on UNIX

If NerveCenter is not running as a UNIX daemon, you can start it manually.

CAUTION

The NerveCenter 5.1 server will not start without a valid license file, as described in *About NerveCenter Licensing in Installing NerveCenter*.

TO START NERVECENTER SERVER ON UNIX

1. Before you can run any NerveCenter application on UNIX, you must set the necessary environments by executing one of the following shell scripts:

   - *install_path*/userfiles/ncenv.sh
   - *install_path*/userfiles/ncenv.csh
   - *install_path*/userfiles/ncenv.ksh

   where *install_path* is typically /opt/OSInc.

   TIP

   An administrator should add the lines from these files to the login scripts of all users who are going to be using NerveCenter on UNIX.

2. From the *install_path*/bin directory, enter the following command:

   ```
   install_path/bin/ncstart
   ```

   NOTE

   If you want to disable all alarms when you start the server (for example, if you create an alarm that crashes the server) type **ncstart -off**.

TO STOP NERVECENTER SERVER ON UNIX

- From the *install_path*/bin directory, enter the following command:

  ```
  ./ncstop
  ```

For other command line switches, see *Controlling NerveCenter from the Command Line on page 277*.

Running the NerveCenter Server on Windows

During a NerveCenter installation, the NerveCenter Server is installed as a Windows service. During installation, the choice is given to have the NerveCenter Server service start up automatically. If the NerveCenter Server is installed as a service but not as an automatic service, you must start it manually each time.

CAUTION

The NerveCenter 5.1 server will not start without a valid license file, as described in *About NerveCenter Licensing in Installing NerveCenter*.

TO START THE NERVECENTER SERVER AS A SERVICE

Use any of the following methods:

- Choose **Start** > **Programs** > **OpenService NerveCenter** > **NerveCenter Service**.
- Select the NerveCenter Service from Service applet in the Windows Control Panel and choose **Start**.
- From the command line, type:

```
ncstart -s
```

TO STOP RUNNING THE NERVECENTER SERVER AS A SERVICE

Use either of the following methods:

- Select the NerveCenter Service from the Windows Services applet in the Windows Control Panel and choose **Stop**.
- From the command line, type:

```
ncstop -s
```

to stop the NerveCenter Service

or

```
ncstop -All
```

to stop all running NerveCenter applications.

Troubleshooting: Running the NerveCenter Server

This section contains some common problems users may have when running the NerveCenter Server.

UNIX will not start the NerveCenter Server

Problem: The environment variables are not set.

Solution: Execute one of the following shell scripts:

- *install_path*/userfiles/ncenv.sh
- *install_path*/userfiles/ncenv.csh
- *install_path*/userfiles/ncenv.ksh

where *install_path* is typically /opt/OSInc**.**

An alarm causes the NerveCenter Server to crash every time I start it

Problem: Any alarm that is enabled when a NerveCenter Server stops is still enabled when the NerveCenter Server starts again.

Solution: Start the NerveCenter Server with all its alarms turned off by typing at the command prompt **ncserver -off &**.

The NerveCenter Server does not start on Windows

Problem: The NerveCenter Server was installed as a manual service.

Solution: Start the NerveCenter Service manually by typing at the command prompt **ncstart -s**.

I need to get a better idea of what the NerveCenter Server is doing on Windows. Can I temporarily run it as a process?

Problem: NerveCenter typically runs as a Windows service, which causes a number of its functions to be hidden to users.

Solution: You can start NerveCenter Server as a Windows process by typing at the command line **ncstart -p**.

Connecting to a NerveCenter Server

5

For you to administer one or more NerveCenter Servers, you will need to make a connection between the NerveCenter Server and a NerveCenter Administrator application.

This chapter includes the following sections:

Section	Description
Starting NerveCenter Administrator on page 42	Explains how to start the NerveCenter Administrator application.
Connecting Administrator to a NerveCenter Server on page 43	Explains how to connect a NerveCenter Administrator to a NerveCenter Server.
Connecting Administrator to Multiple NerveCenter Servers on page 44	Explains the issues involved in connecting a single NerveCenter Administrator to more than one NerveCenter Server.
Deleting a Name from the Administrator Server List on page 45	Explains how to delete a misspelled or old server name from the Administrator's server list.
Viewing NerveCenter Server Status on page 46	Explains how to view the Server Status window to view basic information concerning the active NerveCenter Server.
Disconnecting the Administrator from a NerveCenter Server on page 49	Explains how to disconnect the NerveCenter Administrator from a NerveCenter Server without closing the Administrator window.
Troubleshooting: Connecting to the NerveCenter Server on page 50	Lists common problems users face when starting the NerveCenter Administrator and connecting to a NerveCenter Server.

Starting NerveCenter Administrator

The NerveCenter Administrator provides an administrator information about one or more active NerveCenter Servers. An administrator also uses the application to configure settings on a NerveCenter Server.

Starting the NerveCenter Administrator differs depending on the operating system:

- *Starting the NerveCenter Administrator on UNIX on page 42*
- *Starting the NerveCenter Administrator on Windows on page 42*

Starting the NerveCenter Administrator on UNIX

TO START THE NERVECENTER ADMINISTRATOR ON UNIX

1. Open a terminal window.
2. Type the following command:

   ```
   ncadmin &
   ```

   If you receive an error message, it is possible the necessary environment variables have not been set. See *Running the NerveCenter Server on UNIX on page 38*.

   A NerveCenter Administrator window opens.

Starting the NerveCenter Administrator on Windows

TO START THE NERVECENTER ADMINISTRATOR ON WINDOWS

1. From the Windows task bar, select **Start** > **Programs** > **OpenService NerveCenter** > **Administrator**.

   A NerveCenter Administrator window opens.

   TIP
   It is also possible to start the NerveCenter Administrator from a command prompt (or **Start** > **Run**) by typing **ncadmin**.

Connecting Administrator to a NerveCenter Server

Before you can use the NerveCenter Administrator, you must connect to a NerveCenter Server. The NerveCenter Server provides the functionality required to manage your network. Also, it gives the NerveCenter Administrator access to the settings found in the NerveCenter database.

You can connect your NerveCenter Administrator to more than one server at one time. However, only one NerveCenter Server can be the *active server* at a time. The active server determines which NerveCenter database is used when you are configuring settings.

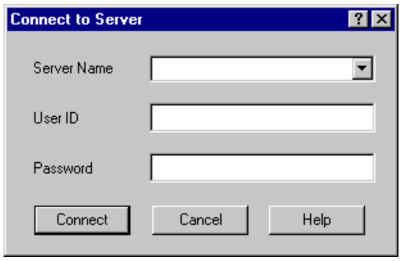
To connect NerveCenter Administrator to a server, you must be an administrator or a member of a NerveCenter group on the server:

- If using Windows, you must be a member of the NerveCenter Admins group or be logged in as an administrator.
- If using UNIX, you must be a member of the ncadmins group or be logged in as root.

TO CONNECT A NERVECENTER ADMINISTRATOR TO A NerveCenter SERVER

1. Open NerveCenter Administrator. See *Starting NerveCenter Administrator on page 42*.
2. From the **Server** menu, select **Connect Server**.

   The Connect to Server window is displayed.



3. Do one of the following:

   a. If this is the first time connecting to the NerveCenter Server, type in the Server Name field the hostname or IP address of the machine on which the NerveCenter Server is running.

   b. If you have connected to the server before, select the name or IP address from the Server Name list.

The first time you connect to a server, the list is empty. After that, it contains a list of the machines to which you've connected, or attempted to connect, in the past. The list will not display the names of machines to which this NerveCenter Administrator is already connected. (For information on removing an entry from the Server Name list, see the section *Deleting a Name from the Administrator Server List on page 45*.)

4. Type a user name and password in the **User ID** and **Password** fields.

   You must enter a user name and password. The user whose name you enter here must be a member of the NerveCenter Users or NerveCenter Admins group (Windows) or the ncusers or ncadmins group (UNIX).

5. Select **Connect**.

   If the machine to which you try to connect is not running a NerveCenter Server, you will see the message: **The server did not respond.** You must first start the NerveCenter Server and then try to connect again. See *Running the NerveCenter Server on page 37* for more details.

When the NerveCenter Administrator connects to a NerveCenter Server, the Administrator window for that server appears.

Connecting Administrator to Multiple NerveCenter Servers

It is possible to use a NerveCenter Administrator to connect to more than one NerveCenter Server at the same time. However, only one NerveCenter Server can be the active server at a time. The active server determines which NerveCenter database is used when you are configuring settings.

To connect a NerveCenter Administrator to more than one NerveCenter Server, see *Connecting Administrator to a NerveCenter Server on page 43*. NerveCenter Administrator displays a window for each connected NerveCenter Server. The name of the active server always appears in the title bar of the main NerveCenter Administrator window.

Deleting a Name from the Administrator Server List

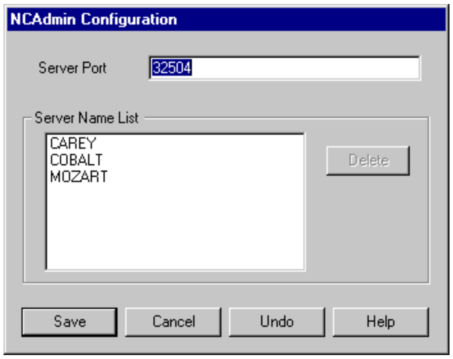
NerveCenter maintains a list of servers that the NerveCenter Administrator has attempted to connect to in the past. This list is used in the Connect to Server window, which you use to connect the administrator to a server.

As helpful as this list is, there may be times when you need to delete a server from this list. It may be the name of a server that you will never connect to again. It could also be the misspelled name of a server you were unable to connect to because of the misspelling.

To delete a server from the Administrator's server name list

1. Open NerveCenter Administrator. See *Starting NerveCenter Administrator on page 42*.
2. From the **Admin** menu, select **Configuration**.

   The NCAdmin Configuration window is displayed.

   
   NCAdmin Configuration

   Server Port: 32504

   Server Name List: CAREY, COBALT, MOZART

   The Server Name List contains the names of all the machines this NerveCenter Administrator has attempted to connect to in the past.
3. In the Server Name List, highlight the machine name that you wish to delete.
4. Select **Delete**.

   The machine name is removed from the Server Name List.
5. Select **Save**.

The server's name is permanently deleted from the Server Name List for this NerveCenter Administrator.

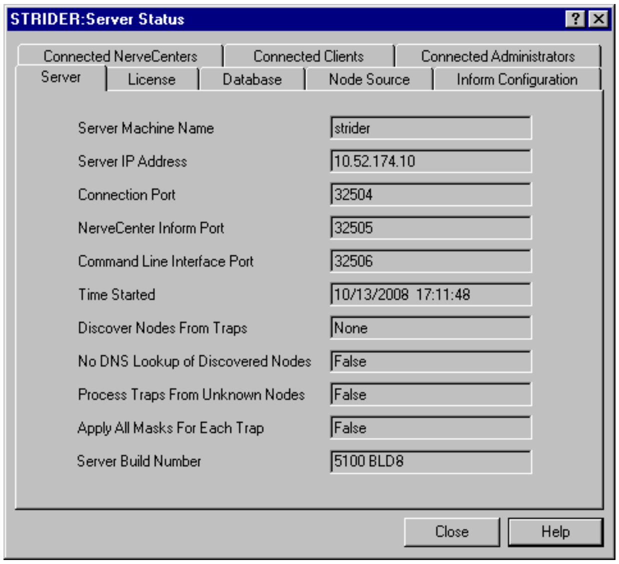
Viewing NerveCenter Server Status

The Server Status window of NerveCenter Administrator and Client allow you to see current information pertaining to an active NerveCenter Server.

To view information about a NerveCenter Server

1. Open NerveCenter Administrator and connect to a NerveCenter Server See *Connecting Administrator to a NerveCenter Server on page 43*.
2. From the **Server** menu, select **Server Status**.

   NerveCenter displays the Server Status window for the active NerveCenter Server.



Server Machine Name	strider
Server IP Address	10.52.174.10
Connection Port	32504
NerveCenter Inform Port	32505
Command Line Interface Port	32506
Time Started	10/13/2008 17:11:48
Discover Nodes From Traps	None
No DNS Lookup of Discovered Nodes	False
Process Traps From Unknown Nodes	False
Apply All Masks For Each Trap	False
Server Build Number	5100 BLD8

3. To close the window, select **Close**.

The Server tab presents information about the machine the NerveCenter server is running on, the communication ports being used by NerveCenter, and the server’s node-discovery settings. *Table 5-1* describes the information on the Server page:

TABLE 5-1. Fields on Server Tab

Label	Explanation
Server Machine Name	The name of the host running the NerveCenter Server.
Server IP Address	The IP address of the server.
Connection Port	The port used by the server to communicate with the client.
NerveCenter Inform Port	The port used by the server to receive Inform actions from other NerveCenter servers.
Command Line Interface Port	The port number on the server used for the command line interface.
Time Started	The time that the server was started (in the server’s time zone).
Discover Nodes from Traps	How nodes are to be discovered. If NerveCenter is set up to discover nodes, it adds nodes to the database based on this setting: ◆ **None** - Never add an unknown node (NerveCenter discovery is disabled). ◆ **All** - Add all unknown nodes. ◆ **IP Filter** - Add the node if its IP address matches the IP filter criteria specified in NerveCenter Administrator.
No DNS Lookup of Discovered Nodes	**False**, the default, indicates that NerveCenter will attempt a DNS lookup for any IP address discovered from a trap. **True** indicates that NerveCenter will not attempt a DNS lookup of any node discovered by a trap. Nodes are added to NerveCenter as an IP address. **Note:** No DNS Lookup of Discovered Nodes is only valid if Discover Nodes from Traps is selected.
Process Traps from Unknown Nodes	**True** indicates that NerveCenter processes traps from all nodes, regardless of filters imposed by NerveCenter or a network management platform. If **False**, NerveCenter discards traps coming from nodes outside your filters. This feature does not change the effect your filters have on discovery. While traps from any node can be processed, nodes are added to the NerveCenter database only if they meet your filter criteria.

TABLE 5-1. Fields on Server Tab (Continued)

Label	Explanation
Apply All Masks for Each Trap	**True** indicates that NerveCenter processes every incoming SNMP trap against all defined trap masks that are currently enabled. A mask processes traps even when its associated alarm is turned off or is not in a state that can be transitioned by the mask's trigger.
Server Build Number	The NerveCenter software version running on the server.

The other tabs in Server Status window display information about the active NerveCenter Server. *Table 5-2* describes the other Server Status Tabs.

TABLE 5-2. Server Status Tabs

Tab	Description
License	Displays license information for the active NerveCenter Server.
Database	Displays information about the database this NerveCenter Server is using. Includes statistics on its current behavior models.
Node Source	Displays the node filters as well as node source, if applicable, this NerveCenter Server is using to maintain its node list.
Inform Configuration	Displays the settings involved in forwarding event information to one or more inform recipients.
Connected NerveCenters	Displays information about each NerveCenter Server that may potentially send an inform packet to the active NerveCenter Server.
Connected Clients	Displays information about each host running a NerveCenter Client connected to the active NerveCenter Server.
Connected Administrators	Displays information about each host running a NerveCenter Administrator connected to the active NerveCenter Server.

\

Disconnecting the Administrator from a NerveCenter Server

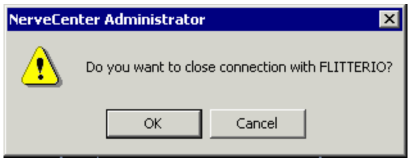
When you no longer need to configure settings in NerveCenter Administrator, you can disconnect from a NerveCenter Server. If you are connected

When you close a NerveCenter Administrator, all connections to NerveCenter servers are broken. However, you may also want to disconnect the administrator from a server without stopping a NerveCenter Administrator.

To disconnect a NerveCenter Administrator from a server

1. In NerveCenter Administrator, select the window of the server from which you want to disconnect.
2. From the **Server** menu, choose **Disconnect Server**.

   A warning message appears asking if you are sure you want to disconnect from the active server.


NerveCenter Administrator

Do you want to close connection with FLITTERIO?

3. Select **OK**.

NerveCenter Administrator remains open but is no longer connected to the NerveCenter Server.

Troubleshooting: Connecting to the NerveCenter Server

The following list contains some common problems users have when starting NerveCenter Administrator and connecting to the NerveCenter Server.

UNIX will not start the NerveCenter Administrator

Problem: The environment variables are not set.

Solution: Execute one of the following shell scripts:

- /opt/OSInc/userfiles/ncenv.sh
- /opt/OSInc/userfiles/ncenv.csh
- /opt/OSInc/userfiles/ncenv.ksh

See *Running the NerveCenter Server on UNIX on page 38*.

While trying to connect to a NerveCenter Server I get the message: The server did not respond

Problem: The NerveCenter Server is not currently running.

Solution: Start the NerveCenter Server and then connect.

See *Running the NerveCenter Server on UNIX on page 38* or *Running the NerveCenter Server on Windows on page 39*.

I misspelled a server name while trying to connect and now the misspelled name appears in the NerveCenter Administrator Server Name list

Problem: NerveCenter Administrator remembers the name of every server you attempted to connect to.

Solution: Remove the misspelled name by deleting it from the Administrator's configuration window.

See *Deleting a Name from the Administrator Server List on page 45*.

Managing the NerveCenter Server

6

NerveCenter is based on a client/server architecture. This means the user views information and interacts with NerveCenter through client applications while most of NerveCenter's actions are carried out by the NerveCenter Server.

Much of the information NerveCenter needs to detect and correlate events are provided when behavior models are created. However, a NerveCenter administrator must configure and maintain certain foundational settings.

NOTE

To change the default port settings used by NerveCenter for various communications, see *Managing NerveCenter Port Settings on page 73*.

This chapter includes the following sections:

Section	Description
Managing the NerveCenter Trap Source on page 52	Explains how to determine and change the trap source, or trap engine, you want to use for capturing SNMP traps.
Reconfiguring a NerveCenter Server from the Command Line on page 61	Explains how to change settings to a NerveCenter Server by importing an .ini file.
Troubleshooting: Managing the NerveCenter Server on page 71	Lists common problems users face when managing the NerveCenter Server.

Managing the NerveCenter Trap Source

During installation, you indicated which trap source, or trap engine, you wanted to use for capturing SNMP traps. You can later change this trap source. Keep in mind that neither MSTrap nor OVTrapD support SNMPv3.

See the following sections for more information:

- *Trap Source Overview on page 52*
- *Changing the SNMP Trap Source on page 55*

Trap Source Overview

During installation, you indicated whether NerveCenter should be configured to support SNMPv3. Depending on your answer, your platform, and whether OpenView Network Node Manager is coresident on your system, your trap source is one of the following:

TABLE 6-1. NerveCenter SNMPv3 Trap Sources

Platform	OpenView Coresident	OpenView Not Coresident
Windows	NerveCenter or OVTrapD	NerveCenter MSTrap
UNIX	NerveCenter or OVTrapD	NerveCenter

When NerveCenter is your trap source, NerveCenter uses the port specified on the Ports tab for capturing traps. This value, SNMP Trap port, is 162 by default. If you want NerveCenter to capture traps, this port must be free for NerveCenter to capture the traps.

During installation, if you selected NerveCenter as your SNMP Trap Source, you were asked to disable MSTrap or OVTrapD service on your system so that the trap port (162) would be free for NerveCenter to use.

After installation, you can change the trap source in NerveCenter Administrator. Keep in mind that neither MSTrap nor OVTrapD support SNMPv3.

Following are the types of traps supported by the trap engines:

TABLE 6-2. SNMP Trap Support

Trap Source	SNMP Version			Name
	v1	v2	v3	
NerveCenter	✓	✓	✓	—
OpenView	✓	✓	✗	ovtrapd
Microsoft	✓	✓	✗	SNMP Trap Service

NerveCenter Trap Utilities

NerveCenter includes the following trap-related utilities:

- **Trapgen** — Generates a trap or inform on a specified node.
- **Traprcv** — Receives traps and shows the traps in standard output.

The following sections briefly describe each utility; see *Appendix C, Controlling NerveCenter from the Command Line* for more information.

Trapgen

TrapGen is a standalone utility that allows you to send an SNMP trap or inform to a particular device. From a command line or shell prompt, you can set the generic and specific trap numbers, provide the sender and recipient IP addresses, include an enterprise identifier, and specify the variable bindings.

Generating a trap is useful for testing trap masks and behavior models. When you send an SNMP trap to a NerveCenter Server, any trap mask configured to detect the specified trap values fires its trigger. If your trap mask and alarm are working properly, you see the alarm listed in your console.

Traprcv

The traprcv command displays the SNMP Trap messages received by the NerveCenter Trap service. This utility can be useful when debugging behavior models. When you run the traprcv command from a command line or shell prompt, any trap you receive is shown on the screen.

Traprcv can receive the following traps and informs:

- SNMPv1 Traps
- SNMPv2c Traps
- SNMPv2c Informs
- SNMPv3 Traps
- SNMPv3 Informs

Changing the SNMP Trap Source

When NerveCenter is your trap source, NerveCenter uses the port specified on the Ports tab for capturing traps. This value, SNMP Trap port, is 162 by default. If you want NerveCenter to capture traps, this port must be free for NerveCenter to capture the traps. This may require that you shut down MS Trap service and/or OVTrapD.

Follow the steps below to change the trap source used to process traps. The sequence is important because the current trap source has to release the port before the new trap source can use that port.

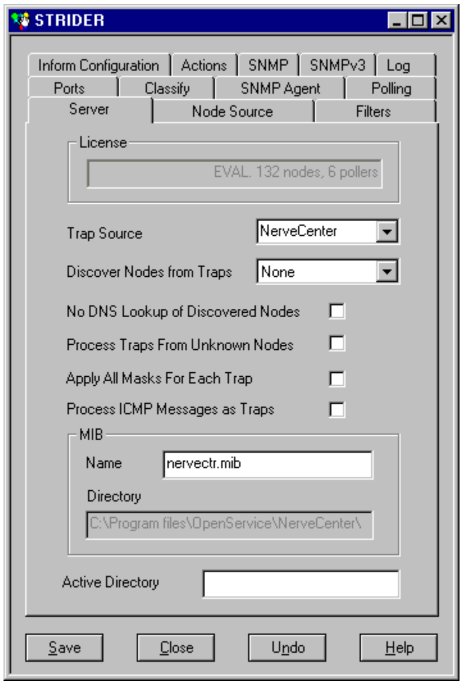
TO CHANGE THE NERVECENTER TRAP SOURCE

1. Open NerveCenter Administrator and connect to the appropriate Server.

   See *Connecting Administrator to a NerveCenter Server on page 43*.

2. Select the **Server** tab.

   The Server page is displayed.



License: EVAL. 132 nodes, 6 pollers

Trap Source: NerveCenter

Discover Nodes from Traps: None

No DNS Lookup of Discovered Nodes ☐

Process Traps From Unknown Nodes ☐

Apply All Masks For Each Trap ☐

Process ICMP Messages as Traps ☐

MIB

Name: nervectr.mib

Directory: C:\Program files\OpenService\NerveCenter\

Active Directory:

3. Select the trap engine you want to use from the Trap Source drop-down listbox.

4. Select **Save**.
5. Shut down the NerveCenter Server.
6. If applicable, stop the old trap service and then start the new one.

   Refer to *Table 6-3* for details.
7. Restart the NerveCenter Server.

Table 6-3 summarizes what to do when changing the trap source.

TABLE 6-3. Changing the Trap Source

Switching From	Switching To	Action To Take...
OVTrapD or MS Trap	NerveCenter	Shut down OVTrapD and MS Trap service. If both are on your system, shut down both. Disable the MS Trap Service. If SNMP Trap port is other than 162 in NerveCenter, you don’t need to shut down OVTrapd or MSTrap service.
NerveCenter	OVTrapD	Start OVTrapD.
NerveCenter	MS Trap	Start MS Trap service. Disable the NerveCenter SNMP Trap service.
OVTrapD	MS Trap	Stop OVTrapD. Start MS Trap service.
MS Trap	OVTrapD	Stop MS Trap service. Start OVTrapD.

To change from the NerveCenter Trap Service to MSTRAP

1. Select MSTRAP as your trap source.

   a. Open NerveCenter Administrator and connect to the appropriate Server.

   b. Select the **Server** tab.

      The Server tab displays.

   c. Select **MSTRAP** from the Trap Source drop-down listbox.

   d. Select **Save**.

      A dialog box displays stating Data saved successfully.

2. Disconnect from the Server and shut down NerveCenter Administrator.

3. Stop the following services in the order given:

   a. NerveCenter service

   b. NerveCenter SNMP service

4. Set Startup type of the SNMP Trap Service to Automatic.

   Follow the instructions of *To disable the MS Trap service on page 61* but select **Automatic** instead of **Disabled**.

5. Start the following service in the order given:

   a. SNMP Trap Service service

   b. NerveCenter SNMP service

   c. NerveCenter service

If you have snmputil installed, you can test that MS Trap service is receiving traps.

To test whether the MS Trap service is receiving traps

1. Run snmputil trap at a command prompt.

   The utility returns snmputil: listening for traps...

2. Enter the following command:

```
trapgen -v1 your_ip-address 1.3.6.1.4.78 your_ip-address 6 101 132
```

snmputil should return the following text:

```
Incoming Trap:
generic       = 6
specific      = 101
enterprise    = .iso.org.dod.internet.private.enterprises.78
agent         = your ip-address
source IP     = your ip-address
community     = public
```

If the AllTraps_LogToFile alarm is enabled, the alltraps.log should contain the following lines.

```
Time=10/22/2002 13:11:02 Tue; LogId=1; DestStateSev=Normal;
NodePropertyGroup=NC
DefaultGroup; NodeName=mycomputer;
AlarmName=AllTraps_LogToFile;
OrigState=Ground;
TriggerName=allTraps; DestState=Logging; TrapPduTime=132;
TrapPduGenericNumber=6
; TrapPduEnterprise=1.3.6.1.4.1.78;
TrapPduSpecificNumber=101;
TriggerInstance=;
TriggerBaseObject=
```

To change from the MSTRAP Service to NerveCenter Trap Service

1. Select NerveCenter as your trap source.
    a. Open NerveCenter Administrator and connect to the appropriate Server.

       For instructions, see *Connecting to a NerveCenter Server on page 41*.
    b. Select the **Server** tab.

       The Server tab displays.
    c. Select NerveCenter from the Trap Source drop-down listbox.
    d. Select **Save**.

       A dialog box displays stating **Data saved successfully**.
2. Disconnect from the Server and shut down NerveCenter Administrator.

   For instructions, see *Disconnecting the Administrator from a NerveCenter Server on page 49*.
3. Stop the following services in the order given:
    a. NerveCenter service
    b. NerveCenter SNMP service
    c. SNMP Trap Service service
4. Set Startup type of the SNMP Trap Service to Disabled.

   Follow the instructions of *To disable the MS Trap service on page 61*.
5. Start the following services in the order given:
    a. NerveCenter SNMP service
    b. NerveCenter service

You can test that the NerveCenter Trap service is running.

TO TEST WHETHER THE NERVECENTER TRAP SERVICE IS RECEIVING TRAPS

1. Run traprcv in a command window.

   The utility returns Waiting for traps.

2. Enter the following command:

```
trapgen -v1 your_ip-address 1.3.6.1.4.78 your_ip-address 6 101 132
```

traprcv should return the following text:

```
Received SNMPv1 Trap:
Community: public
From: your ip-address:2057
Enterprise: enterprises.7
Agent-addr: your ip-address
Enterprise Specific trap: 101
Time Ticks: 132
```

If the AllTraps_LogToFile alarm is enabled, the alltraps.log should contain the following lines.

```
Time=10/22/2002 13:38:02 Tue; LogId=1; DestStateSev=Normal;
NodePropertyGroup=NC
DefaultGroup; NodeName=mycomputer;
AlarmName=AllTraps_LogToFile;
OrigState=Ground;
TriggerName=allTraps; DestState=Logging; TrapPduTime=132;
TrapPduGenericNumber=6
; TrapPduEnterprise=1.3.6.1.4.1.78;
TrapPduSpecificNumber=101;
TriggerInstance=;
TriggerBaseObject=
```

TO DISABLE THE MS TRAP SERVICE

1. Open the Services control panel by selecting the following from the **Start** menu:

   Settings > **Control Panel** > **Administrative Tools** > **Services**
2. Right-click on the **SNMP Trap Service** and select **Properties** from the menu.
3. From the Startup Type listbox, select **Disabled**.
4. Select **OK**.

Reconfiguring a NerveCenter Server from the Command Line

An alternative to using the NerveCenter Administrator to configure a NerveCenter Server is to use the utility ImportUtil. ImportUtil is a utility that does the following from the command line:

- Imports behavior models and nodes to a NerveCenter Server.
- Imports configuration settings to a NerveCenter Server.

Using this utility allows you to reconfigure a setting on more than one NerveCenter Server at a time by changing one file and importing it to all the relevant servers. *Figure 6-1* shows ImportUtil along with the other NerveCenter components.

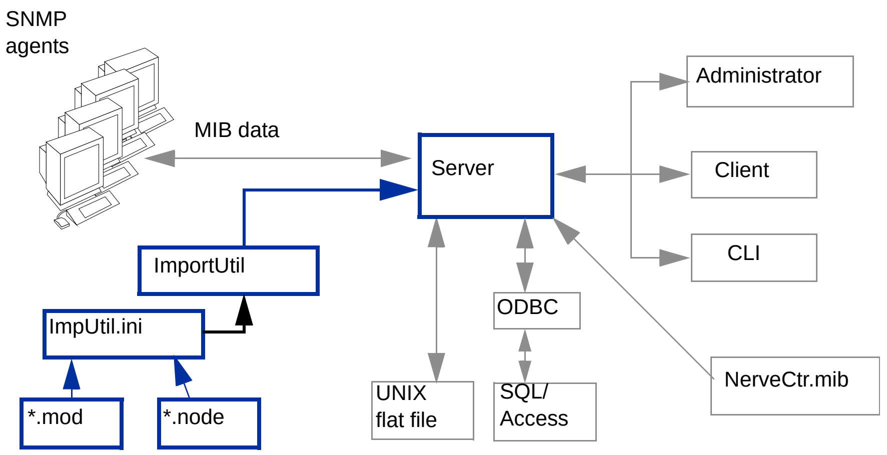


FIGURE 6-1. NerveCenter ImportUtil Utility

ImportUtil imports the data referenced in the editable file ImpUtil.ini, which is located in the *installation/*Sms directory (Windows) or the *installation/*userfiles directory (UNIX).

You can edit NerveCenter configuration settings in ImpUtil.ini as well as specify the behavior models and nodes you want to import. You must have the proper *.mod or *.node file for the behavior models or nodes you want to import. When you have made the changes you want to ImpUtil.ini, run ImportUtil to import the specified information or files.

See *Appendix C, Controlling NerveCenter from the Command Line* for more information.

TO RECONFIGURE A NERVECENTER SERVER FROM THE COMMAND LINE

1. Find the file imputil.ini.

   - In a typical NerveCenter installation on Windows, this file can be found in the folder **Sms**, in the NerveCenter folder.
   - In a typical NerveCenter installation on UNIX, this file can be found in the directory *install_path*/userfiles, where *install_path* is typically /opt/OSInc.

   The file imputil.ini is made of a number of sections that include a section header and keys.

2. Before making any changes, create a backup copy of the file imputil.ini.

   CAUTION
   If you do not edit imputil.ini correctly, you can misconfigure or corrupt your NerveCenter Server. You may want to test your changes to imputil.ini in a test environment before running importutil in an production environment.

   You will not be able to restore the original imputil.ini after making changes to the file, unless you first make a backup copy.

3. Delete all but the relevant sections to be changed.

   Table 6-4 on page 64 describes all sections in imputil.ini. All sections in the file are optional. If you remove a section, including the section header and keys, ImportUtil does not change or delete any values in the NerveCenter configuration settings for that key.

   For example, if you are changing a value found only in the section **[CONFIG_SERVER]**, you delete all sections except the section header and the values in the **[CONFIG_SERVER]** section. ImportUtil will only change the values pertaining to that section.

4. Within the remaining sections, delete everything but the section headers and the relevant keys.

   Any new values left in imputil.ini will overwrite the old values. To avoid having placeholders overwrite legitimate values, delete any unnecessary keys before running ImportUtil.

   For example, if within the **[CONFIG_SERVER]** section you only want to change the value of the key InformNCListenPort, delete all but the following:

   ```
   [CONFIG_SERVER]
   InformNCListenPort = port
   ```

   ⚠ **CAUTION**
   If you are configuring the **[CONFIG_PLATFORM_NETNODENOTIFY]** or **[CONFIG_PLATFORM_MAPSUBNETS]** sections, you must include all values, including old values. ImportUtil deletes values from the NerveCenter configuration settings that are not included in these sections. Read the comments before each section in this file for more information.

5. Change the values by replacing the placeholders after the equal sign (=) with valid values.

   Unless otherwise noted, you may not leave the value after a key blank.

   For example, if you want to change the value of the key InformNCListenPort to 6024, change the file to read:

   ```
   [CONFIG_SERVER]
   InformNCListenPort = 6024
   ```

   Table 6-4 describes all imputil.ini sections.

6. Save the changed file.

7. While the NerveCenter Server is running, run the utility ImportUtil from a UNIX shell or a Windows command prompt by typing:

   ```
   importutil imputil.ini
   ```

   ⚠ **CAUTION**
   You must either be in the same directory as the imputil.ini file or include the full pathname of the imputil.ini file.

NerveCenter notifies you upon successful completion of the reconfiguration.

TABLE 6-4. imputil.ini Value Descriptions

imputil.ini value pair	Description
[CONFIG_PLATFORM]	
This section configures integrations with network management platforms.	
To remove a value, leave the value after the equal sign (=) blank. For example: MapHost=	
Note: Except for OpCHost, cross-references in this section refer to OpenView integrations. Review the chapter in *Integrating NerveCenter with a Network Management Platform* appropriate for your network managment platform.	
MapHost = host/port/resyncparentrate	These are the hostname, port, and resync Parent rate defined on the Node source tab. For details, see *Identifying Parent-Child Relationships in Integrating NerveCenter with a Network Management Platform*
Method = Auto | Manual	When the method is set to Automatic, NerveCenter ignores your address filters. It looks at the IP address and mask for each network interface card on that server and uses them to calculate the subnet address and mask. Enter Manual if you want NerveCenter to use your address filters as entered. For details see *Filtering Nodes by IP Address on page 99*
WantedCapabilities = cap1,cap2,...	You can monitor nodes with particular capabilities, which your network management system typically assigns to a node to determine the applicable activities such as isRouter, isHub, and isIP. Filtering by capabilities is available only when your network management platform has assigned specific capabilities to a node. For details, see *Filtering Using Node Capabilities on page 97*.
SystemObjectIds = id1,id2,...	NerveCenter allows you to monitor managed nodes according to their particular system object identifiers (OIDs). For details, see *Filtering Using a Node System Object Identifier on page 98*.
MaxQueueDepth = maximum queue depth	The number of Informs NerveCenter keeps if connection with the network managment platform goes down. For details, see *Configuring the Inform Queue Depth in Integrating NerveCenter with a Network Management Platform*
AckInforms = 0/1	If you want NerveCenter to keep track of informs until the platform adapter acknowledges their receipt, you must enable the inform acknowledgement feature by setting AckInforms = 1. For details, see *Saving NerveCenter Informs Until Acknowledgement in Integrating NerveCenter with a Network Management Platform*.
NoFiltersOnImport = 0/1	If you have a node list you want to import of known hosts you want to monitor, you may not want to apply the IP and Hostname filters. Enter 0 for false and 1 for true. For details, see *Enabling and Disabling IP and Hostname Filters on page 105*.

TABLE 6-4. imputil.ini Value Descriptions (Continued)

imputil.ini value pair	Description
HostnameFilters = hostname1 *hostname2 . . .	Besides IP addresses, you can also filter by hostnames. For details, see *Filtering Nodes by Hostnames on page 104*.
[CONFIG_PLATFORM_MAPSUBNETS] This section configures the IP filtering for node lists imported from a network management platform. If the method in [CONFIG_PLATFORM] is AUTO, the entries in this section are ignored. **Caution:** You must include all values from this section, including old values. ImportUtil deletes values from the NerveCenter configuration settings that are not included in these sections. **Note:** If you want to apply filters to a list of nodes you are importing, first run importutil to change the filters. Then run importutil again to import the nodes.	

MapSubNet1 = Address;Mask;Exclusion list
MapSubNet2 = Address;Mask;Exclusion list
	You can filter nodes by subnet. For details, see *Filtering Nodes by IP Address on page 99*.

[CONFIG_PLATFORM_NETNODENOTIFY] This section configures what information NerveCenter sends to the Network Managment Platform. **Caution:** You must include all values from this section, including old values. ImportUtil deletes values from the NerveCenter configuration settings that are not included in these sections. **Note:** Cross-references in this section refer to integrating with HP OpenView. Review the chapter in *Integrating NerveCenter with a Network Management Platform* appropriate for your network managment platform.	
host/port/type = EVENT_ONLY | EVENT_AND_SYMBOL | SYMBOL_ONLY [Min Severity] [Properties]	The type of information sent to the network managment platform. For details about these values, see *Sending NerveCenter Informs to OpenView in Integrating NerveCenter with a Network Management Platform*. **Note:** Type must be either OVPA or PA_SVR
[CONFIG_SERVER] This section configures the NerveCenter Server.	

TABLE 6-4. imputil.ini Value Descriptions (Continued)

imputil.ini value pair	Description
ServerConnPort = port	The NerveCenter Server communicates with other applications, such as NerveCenter Client and NerveCenter Administrator, on a special connection port. For details, see *Configuring the NerveCenter Server Connection Port on page 74*. If you change a NerveCenter Server's communication port number, be sure all applications, such as NerveCenter Administrator, have a matching port number. See *Changing the NerveCenter Administrator Server Port on page 76* For details.
CLIConnPort = port	The port number you want the NerveCenter Server to use for client/server communication. For details, see *Changing the NerveCenter Client Server Port on page 77*.
InformNCListenPort = port	Before a NerveCenter Server can receive an Inform, it must be configured to have a listening port number. For details, see *Configuring NerveCenter to Receive Inform Actions on page 78*.
NCMibName = MibFileName	The filename of the mib file you want to import into NerveCenter. You enter the path in [CONFIG_SERVER_PATH].
EnableDiscovery = TRUE | FALSE	If true, NerveCenter adds nodes discovered by traps to the database. For details, see *Adding Nodes Discovered from Traps on page 117*.
DiscoverNodesFromTraps = All | Filter | None	For details, see *Populating Using the IPSweep Behavior Model on page 112*.
OVTrapperPath = MSTRAP | OVTRAPD | NCNOMSTRAP | NCNOOVTRAP | ALWAYS | NCNEVER | NCALWAYS	**Windows**: ◆ MSTRAP - traps are received from Microsoft SNMP Trap Service. ◆ OVTRAPD - traps are received via ovtrapper from HP OpenView. ◆ NCNOMSTRAP - traps are received via the NerveCenter SNMP Trap agent. Use if NerveCenter is *not* co-resident with HP OpenView. ◆ NCNOOVTRAP - traps are received via the NerveCenter SNMP Trap agent. Use if NerveCenter is co-resident with HP OpenView. **UNIX**: ◆ ALWAYS - traps are received via ovtrapper from HP OpenView ◆ NCNEVER - traps are received via the NerveCenter SNMP Trap agent. Use if NerveCenter is *not* co-resident with HP OpenView. ◆ NCALWAYS - traps are received via the NerveCenter SNMP Trap agent. Use if NerveCenter is co-resident with HP OpenView. For details, see *The OpenView Platform Adapter in Integrating NerveCenter with a Network Management Platform*.

TABLE 6-4. imputil.ini Value Descriptions (Continued)

imputil.ini value pair	Description
useOvTrapd = MSTRAP | OVTRAP	The NerveCenter trap source. For details, see *Managing the NerveCenter Trap Source on page 52*.
ApplyAllMasksForEachTrap = TRUE | FALSE	Set to TRUE if you want NerveCenter to process every incoming SNMP trap against all defined trap masks that are currently enabled. Selecting this option forces masks to process traps regardless of the state of their associated alarms. A mask processes traps even when its associated alarm is turned off or is not in a state that can be transitioned by the mask's trigger. There are times, however, when masks do not process traps though this option may be checked. A mask processes traps only when the following conditions are met: ◆ The mask is turned on. ◆ The incoming trap matches the generic, specific, or enterprise OID values defined in the mask.

TABLE 6-4. imputil.ini Value Descriptions (Continued)

imputil.ini value pair	Description
[CONFIG_SERVER_PATH] Path information.	
MibDirectory = Location of Mib	The path to the mib file you want to import into NerveCenter. You enter the filename in [CONFIG_SERVER].
LogDirectory = Location of logs	The location of any log created by a Behavior Model or other component of NerveCenter. For details, see *Specifying Settings for Log Management on page 191*.
[CONFIG_SERVER_PAGER] This section configures the paging properties.	
TouchPulse =	Enter the frequency. Default is 1200. For details, see *Setting Dialing Properties for Paging Notification on Windows on page 188*.
DialOutNumber =	The number required to reach an outside line, for example, 9. For details, see *Setting Dialing Properties for Paging Notification on Windows on page 188*.
PauseString =	The characters used to create a pause between dialing and sending data. For details, see *Setting Dialing Properties for Paging Notification on Windows on page 188*.
[CONFIG_SERVER_SNMP] This section configures SNMP settings.	
SnmpRetry = No. of retries	The number of times to reissue unanswered SNMP or ICMP request polls. For details, see *Specifying SNMP Poll Intervals for NerveCenter on page 139*.
SnmpInterval = No. of Intervals	The number of seconds NerveCenter should wait for a reply to a poll before issuing another. For details, see *Specifying SNMP Poll Intervals for NerveCenter on page 139*.
DefaultPort = port no.	The port number NerveCenter uses for SNMP traps. For details, see *Specifying SNMP Ports for NerveCenter on page 81*.
[CONFIG_SERVER_SNMPV3] This section configures the SNMPv3 settings.	
PollUser = poll user	The SNMPv3 user name. For details, see *Configuring SNMPv3 Security Settings on page 147*.
PollContext = poll context	The SNMPv3 context. For details, see *Configuring SNMPv3 Security Settings on page 147*.
AutoClassify = on | off	Whether NerveCenter attempts to classify the SNMP format of discovered nodes. For details, see *Classifying Nodes Automatically on page 135*.

TABLE 6-4. imputil.ini Value Descriptions (Continued)

imputil.ini value pair	Description
AuthPwd = unencrypted (plain text) authentication password (min 8 chars and max 32 chars)	The Authentication password for an SNMPv3 node. For details, see *Configuring SNMPv3 Security Settings on page 147*.
PrivPwd = unencrypted (plain text) privacy password (min 8 chars and max 32 chars)	The Privacy password for an SNMPv3 node. For details, see *Configuring SNMPv3 Security Settings on page 147*.
MaxClassifyVer = v1 | v2c | v3	If autoclassification is enabled, you can limit to what level NerveCenter attempts to classify nodes. For details, see *Setting a Maximum Classify Value on page 137*.
MaxReqPerCycle = Max no. of requests going out of v3opmanager per second	You control SNMPv3 performance by setting a maximum number of requests per processing cycle (approximately one second) for all v3 operations including classification, SNMP test version poll, version change, authentication protocol change, security level change, and initialization requests. For details, see *Setting the Maximum SNMPv3 Requests per Cycle on page 138*.
updateKeys = T | F	To update the passwords on all managed agents, enter T. For details, see *Configuring SNMPv3 Security Settings on page 147*.
[CONFIG_SERVER_LOGS] This section configures the NerveCenter Logs.	
MaxLogEntryAge =	The number of hours you want to keep individual log entries before they are deleted from the rest of the log. For details, see *Specifying Settings for Log Management on page 191*.
MaxLogFileSize =	The size limit, in kilobytes, of the ASCII text file storing the results of the Log to File alarm action. For details, see *Specifying Settings for Log Management on page 191*.
MaxNumDBRecords =	The highest number of records in the database file storing the results of the Log to Database alarm action. For details, see *Specifying Settings for Log Management on page 191*.
LogDelPercentage =	The percentage of the log to clear when the maximum file size or the maximum number of records is reached. For details, see *Specifying Settings for Log Management on page 191*.
MaxQueueDepth =	The highest number of changes that you want queued before saving to the database. For details, see *Specifying Settings for Log Management on page 191*.

TABLE 6-4. imputil.ini Value Descriptions (Continued)

imputil.ini value pair	Description
[CONFIG_SERVER_MSMAIL]	
This section configures Microsoft Mail.	
Profile =	The NerveCenter-designated Exchange profile to point to the machine hosting the Microsoft Exchange Server and the NerveCenter-designated mailbox created in *Configuring NerveCenter to Send Microsoft Mail on page 182*.
Password =	The password to the NerveCenter-designated mailbox created in *Configuring NerveCenter to Send Microsoft Mail on page 182*.
[CONFIG_SERVER_SMTPMAIL]	
This section configures SMTP mail.	
SMTPHost =	The name of the host running your SMTP mail server. For details, see *Specifying an SMTP Server for Mail Notification on page 180*.
[IMPORT_MODEL]	
Use this section to import a NerveCenter Behavior Model.	
File = model_path_and_filename	For details, see *Importing Node, Object, and Behavior Model Files in Designing and Managing Behavior Models*.
[IMPORT_NODE]	
Use this section to import a NerveCenter Node List.	
File = node_path_and_filename	For details, see *Importing Node, Object, and Behavior Model Files in Designing and Managing Behavior Models*.

Troubleshooting: Managing the NerveCenter Server

The following list contains some common problems users may have when managing the NerveCenter Server.

I need to make the same changes to several NerveCenter Servers

Problem: It would be time consuming to configure each NerveCenter Server manually from the NerveCenter Administrator.

Solution: Create one imputil.ini file and import it to all the NerveCenter Servers.

See *Reconfiguring a NerveCenter Server from the Command Line on page 61*.

Importing imputil.ini caused unwanted changes to a NerveCenter Server

Problem: ImportUtil imports all values found in imputil.ini, including the defaults.

Solution: When using ImportUtil, delete all but the relevant keys.

See *Reconfiguring a NerveCenter Server from the Command Line on page 61*.

I cannot log in to a UNIX NerveCenter Server.

Problem: The login, as defined on the UNIX system running NerveCenter server, might not exist or else might be blocked or not associated with the correct groups.

Solution: You need to work with the administrators for the UNIX system. The login needs to be a defined, valid account and must be in the 'ncadmins' or 'ncusers' groups. As well, the issue might lie in the UNIX system's PAM configuration.

See *Chapter 11, Managing Security on UNIX* for more on UNIX logins, groups and PAM.

6 Managing the NerveCenter Server

Managing NerveCenter Port Settings 7

NerveCenter is based on a client/server architecture. This means that while the NerveCenter Server does the work, you make changes to the server using a client application, such as NerveCenter Client and NerveCenter Administrator. For these applications and a NerveCenter Server to communicate, they must use the same port connection. The Port tab enables you to specify various ports used by the NerveCenter Server.

This chapter includes the following sections:

Section	Description
Client and Server Communication Ports on page 74	Explains how to change the port at which NerveCenter Server communicates with its client applications.
Configuring NerveCenter to Receive Inform Actions on page 78	Explains how to configure a NerveCenter Server to receive an inform action from another NerveCenter Server.
Changing the Command Line Interface Port on page 80	Explains how to change the port NerveCenter uses for its command line interface.
Specifying SNMP Ports for NerveCenter on page 81	Describes how to configure the ports NerveCenter uses to receive SNMP traps and send/receive SNMP correspondences.
Troubleshooting: Managing NerveCenter Port Connections on page 83	Provides troubleshooting tips related to ports.

Client and Server Communication Ports

NerveCenter is based on a client/server architecture. This means that while the NerveCenter Server does the work, you make changes to the server using a client application, such as NerveCenter Client and NerveCenter Administrator. For these applications and a NerveCenter Server to communicate, they must use the same port connection.

By default, NerveCenter Server communicate on the special port 32504. If for some reason—such as this port is being used by another application—it becomes necessary to change NerveCenter's communication port, an administrator may do so. However, the administrator must follow *all* of these procedures:

- *Configuring the NerveCenter Server Connection Port on page 74*
- *Changing the NerveCenter Administrator Server Port on page 76*
- *Changing the NerveCenter Client Server Port on page 77*

CAUTION
Any application attempting to communicate with a NerveCenter Server must have a matching communication port number.

Configuring the NerveCenter Server Connection Port

The NerveCenter Server communicates with other applications, such as NerveCenter Client and NerveCenter Administrator, on a special connection port.

By default, this port number is 32504.

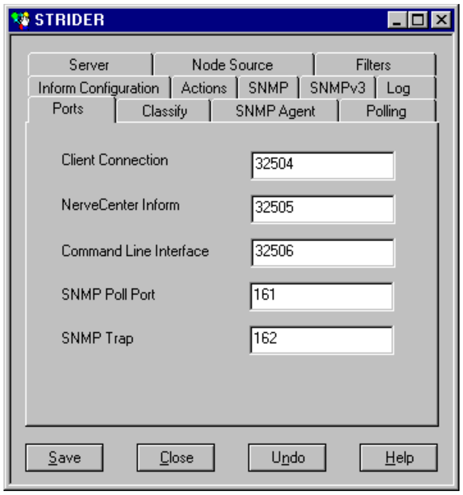
To change the NerveCenter Server connection port

1. Open NerveCenter Administrator and connect to the appropriate Server.

   See *Connecting Administrator to a NerveCenter Server on page 43*.

2. Select the **Ports** tab.

   The Ports tab is displayed.



Client Connection	32504
NerveCenter Inform	32505
Command Line Interface	32506
SNMP Poll Port	161
SNMP Trap	162

3. In the **Client Connection** field, enter the port number you want the NerveCenter Server to use for client/server communication.

   Caution
   If you change the port number here, you must make the same changes to the NerveCenter Client and NerveCenter Administrator.

4. Select **Save**.

5. Stop and start the NerveCenter Server.

   Changes to the communication port will not be complete until the NerveCenter Server is restarted. See *Running the NerveCenter Server on page 37*.

NerveCenter uses this port when connecting with a NerveCenter Administrator or Client.

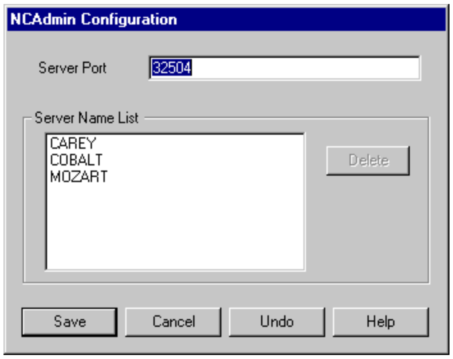
Changing the NerveCenter Administrator Server Port

If you change a NerveCenter Server's communication port number, be sure all applications, such as NerveCenter Administrator, have a matching port number.

TO CHANGE THE NERVECENTER ADMINISTRATOR'S SERVER PORT

1. Open NerveCenter Administrator. See *Starting NerveCenter Administrator on page 42*.
2. From the **Admin** menu, choose **Configuration**.

   The NCAdmin Configuration window appears.

   
   NCAdmin Configuration

   Server Port: 32504

   Server Name List:
   CAREY
   COBALT
   MOZART

   The NCAdmin Configuration window displays the Server port the NerveCenter Administrator application uses to communicate with a NerveCenter Server. It also includes a list of all the servers this application has connected to in the past.
3. In the Server Port field, type the connection port number of the NerveCenter Server you want the NerveCenter Administrator application to connect to.

   CAUTION
   Remember the port number. If you change it here, you must make corresponding changes to the NerveCenter Server.
4. Select **Save**.

The NerveCenter Administrator application will now communicate with the NerveCenter Server at the new port you specified in the Server Port field.

Changing the NerveCenter Client Server Port

If you change a NerveCenter Server’s communication port number, be sure all applications, such as NerveCenter Client, have a matching port number.

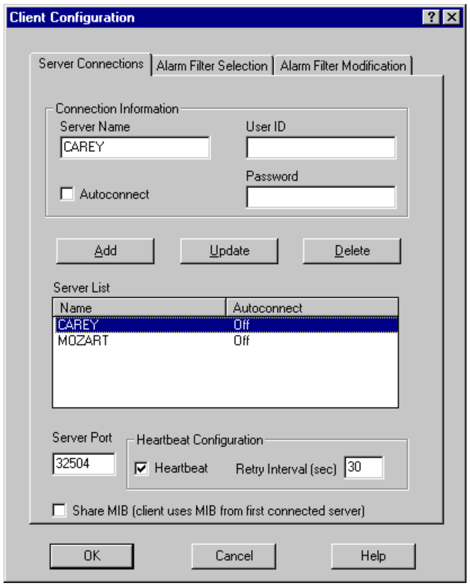
TO CHANGE THE NerveCenter CLIENT’S SERVER PORT

1. Open NerveCenter Client.

   See *Starting the Client in Designing and Managing Behavior Models* for more details.

2. In the **Client** menu, choose **Configuration**.

   The Client Configuration window appears.



Connection Information
Server Name: CAREY
User ID:
Password:
☐ Autoconnect

Server List

Name	Autoconnect
CAREY	Off
MOZART	Off

Server Port: 32504
Heartbeat Configuration
☑ Heartbeat   Retry Interval (sec) 30
☐ Share MIB (client uses MIB from first connected server)

3. Select the **Server Connections** tab.
4. From the server list, select the appropriate server.

   The server’s current port number appears in the Server Port field.

5. In the Server Port field, type the connection port number of the NerveCenter Server you want the NerveCenter Client application to connect to.

CAUTION
Remember the port number. If you change it here, you must make corresponding changes to the NerveCenter Server.

6. Select **Save**.

The NerveCenter Client application will communicate with the NerveCenter Server at the new port you specified in the Server Port field.

Configuring NerveCenter to Receive Inform Actions

Once a NerveCenter behavior model detects a problem, it can notify a network management platform of the problem using the Inform action. This action sends an inform packet to its recipients, indicating the nature of the problem.

NOTE
Although the message that the Inform action sends to its recipients contains the same information as a trap, the message is not sent via UDP. An inform message is sent via TCP to make sure the delivery mechanism is reliable.

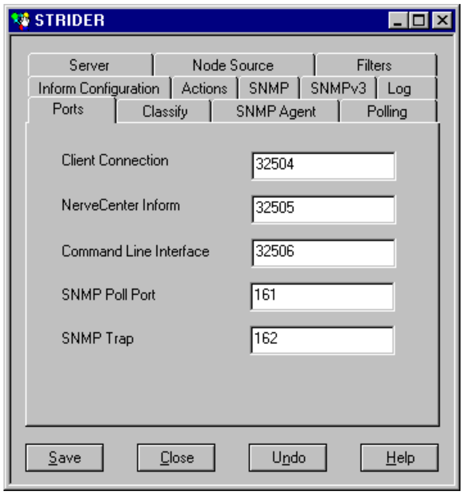
In addition to sending an Inform to network management platforms, a NerveCenter Server can send an Inform to another NerveCenter Server or itself.

Before a NerveCenter Server can receive a trap it must be configured to have a listening port number. By default, this port number is 32505.

TO SPECIFY THE PORT NERVECENTER SERVER USES TO RECEIVE AN INFORM PACKET

1. Open NerveCenter Administrator and connect to an appropriate NerveCenter Server. See *Connecting Administrator to a NerveCenter Server on page 43*.
2. Select the **Ports** tab.

   The Ports tab is displayed.



Client Connection	32504
NerveCenter Inform	32505
Command Line Interface	32506
SNMP Poll Port	161
SNMP Trap	162

3. In the NerveCenter **Inform** field, type the number of the port through which this NerveCenter Server should receive inform packets sent by a NerveCenter Server.
4. Select **Save**.
5. Stop and start the NerveCenter Server.

   Changes to the Inform port will not be complete until the NerveCenter Server is restarted. See *Running the NerveCenter Server on page 37*.

The current NerveCenter Server will receive Inform packets from other NerveCenter Servers at the port specified.

TO VIEW NERVECENTER SERVERS CONFIGURED TO SEND INFORMS TO THE CURRENT SERVER

1. From the **Server** menu in the Administrator or Client, choose **Server Status**.
2. Select the **Connected** NerveCenters tab.

Any NerveCenter Server configured to send an inform packet to the current NerveCenter Server at the port specified in step 3 appears in the Inform NC Name list.

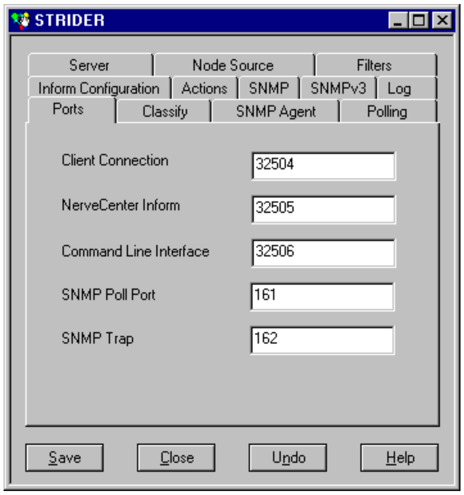
Changing the Command Line Interface Port

You can use NerveCenter's command line interface to delete, list, or set (enable or disable) alarms, trap masks, nodes, and polls from a Windows Command Prompt or a UNIX shell. You can also connect to, display the status of, and disconnect from NerveCenter servers using the CLI. You can issue commands manually or from a script.

TO SPECIFY THE NERVECENTER COMMAND LINE INTERFACE PORT

1. Open NerveCenter Administrator and connect to an appropriate NerveCenter Server. See *Connecting Administrator to a NerveCenter Server on page 43*.
2. Select the **Ports** tab.

   The Ports tab is displayed.



Client Connection: 32504
NerveCenter Inform: 32505
Command Line Interface: 32506
SNMP Poll Port: 161
SNMP Trap: 162

3. In the Command Line Interface field, enter the number of the port through which this NerveCenter Server should receive commands from its Command Line Interface.
4. Select **Save**.

The current NerveCenter Server will receive commands from the port specified.

Specifying SNMP Ports for NerveCenter

NerveCenter has two primary sources of information about network conditions:

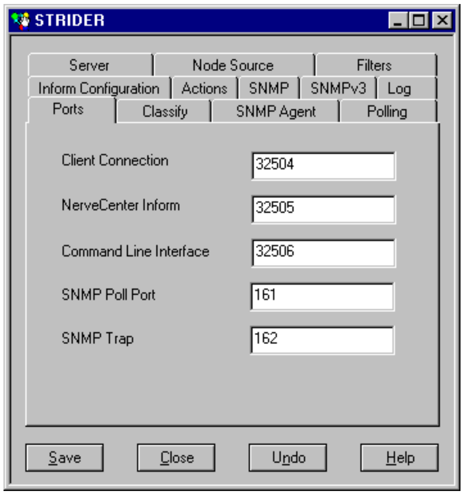
- NerveCenter listens passively for SNMP traps sent by a managed device.
- NerveCenter actively polls the SNMP agents on a managed device.

You can change the ports that NerveCenter uses for receiving traps and sending polls.

TO SPECIFY NERVECENTER SNMP PORT SETTINGS

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server. See *Connecting Administrator to a NerveCenter Server on page 43*.
2. Select the **Ports** tab.

   The Ports tab is displayed.



Client Connection	32504
NerveCenter Inform	32505
Command Line Interface	32506
SNMP Poll Port	161
SNMP Trap	162

3. In the SNMP Poll field, enter the number of the port you want NerveCenter to use to communicate with SNMP agents. This port is used to get or set SNMP information.

   The value entered here specifies the port on the node to which NerveCenter sends SNMP polls. You can change the port for any particular node in the node's definition window in NerveCenter Client.

   LogMatrix recommends that you keep the default port number 161.

4. In the SNMP Trap field, enter the number of the port you want NerveCenter to use for receiving SNMP traps. This setting has no effect if NerveCenter is co-resident with HP OpenView Network Node Manager. OpenView has the trap port and an internal NerveCenter process enables NerveCenter to receive a copy of the trap.

   The default port number is 162.

   NOTE

   You must shut down and restart the NerveCenter Server before the SNMP Trap port change takes effect.

   If you want to capture SNMPv3 traps, the SNMP Trap port must be free for NerveCenter to capture the traps. This requires that you shut down MS Trap service or OVTrapD. If both are on your system, shut down both.

5. Select **Save**.

NerveCenter handles SNMP data according to these settings.

Troubleshooting: Managing NerveCenter Port Connections

The following list contains some common problems users have when managing the NerveCenter Server port connections.

The NerveCenter Server's communication port does not change when I selected Save in NerveCenter Administrator

Problem: NerveCenter must register the changes.

Solution: Stop and start the NerveCenter Server.

See *Running the NerveCenter Server on UNIX on page 38* or *Running the NerveCenter Server on Windows on page 39*.

A NerveCenter Server does not receive Informs sent by another NerveCenter Server

Problem: The inform packet is being sent to an incorrect port.

Solution: Make sure the receiving NerveCenter Server's inform port is the same as the port in the sending NerveCenter's inform configuration.

See *Configuring NerveCenter to Receive Inform Actions on page 78*.

The NerveCenter Server's inform port did not change when I selected Save in NerveCenter Administrator

Problem: NerveCenter must register the changes.

Solution: Stop and start the NerveCenter Server.

See *Running the NerveCenter Server on UNIX on page 38* or *Running the NerveCenter Server on Windows on page 39*.

7 Managing NerveCenter Port Settings

Managing Node Data

8

To detect and correlate network events, NerveCenter must have basic information about each of the managed nodes it is monitoring. It stores this data in a node list in NerveCenter's database. As an administrator you will need to determine which nodes belong in NerveCenter's node list. You will also be responsible for keeping the node list complete and current. This chapter describes the NerveCenter tools you can use to carry out these tasks.

NOTE

NerveCenter can only obtain data from managed nodes running active Simple Network Management Protocol (SNMP) agents.

This chapter includes the following sections:

Section	Description
How NerveCenter Manages Nodes on page 86	Describes the tasks that NerveCenter performs to automate event handling.
The NerveCenter Node List on page 96	Explains that NerveCenter stores basic information about each node it will monitor in a node list in the NerveCenter database.
Filtering Nodes on page 97	Explains how to specify which managed nodes NerveCenter is responsible for monitoring.
Processing Traps from Unknown Nodes on page 107	Explains how to configure NerveCenter to receive and process traps from nodes not included in its node list.
Initially Populating the Node List on page 109	Explains the methods used to populate NerveCenter's node list with initial node data.
Maintaining the Node List on page 115	Explains the various ways nodes can be added to and deleted from NerveCenter's database.
Troubleshooting: Managing Node Data on page 121	Lists common problems NerveCenter users may face when managing node data.

8 Managing Node Data

How NerveCenter Manages Nodes

The following sections describe the tasks that NerveCenter performs to automate event handling:

- *Defining a Set of Nodes on page 86*
- *Detecting Conditions on page 87*
- *Correlating Conditions on page 87*
- *Responding to Conditions on page 92*

Defining a Set of Nodes

NerveCenter can get the list of devices to monitor from a network management platform, discover them on the network, or import this information from another NerveCenter database.

NerveCenter assigns to each managed node a set of *properties*, and these properties determine which behavior models apply to a node. Properties typically describe the type of the device—for example, a router—or are named after objects in the management information base (MIB) used to manage the node.

Once NerveCenter assigns a set of properties to a node, NerveCenter automatically applies to that node all of the models that refer to those properties. If NerveCenter detects that a node has been deleted or that its properties have changed, the product immediately retires or updates the set of models that are actively managing that node. This dynamic process enables NerveCenter to adapt at once to changes in network configuration reported by the management platform or by NerveCenter's own discovery mechanism.

It is also possible to assign properties to nodes manually to further refine the set of models that NerveCenter uses to manage a node. For example, you may want to distinguish a backbone router from a campus router to regulate how much and how often status information is collected.

Detecting Conditions

NerveCenter can collect network and system data from a variety of sources. However, NerveCenter most frequently obtains data from Simple Network Management Protocol (SNMP) agents running on managed nodes. This means that NerveCenter detects most conditions by:

- Receiving and interpreting an SNMP trap
- Polling an SNMP agent for data and analyzing that data

One of the criticisms of SNMP-based enterprise management platforms over the years has been that, because SNMP trap delivery is unreliable, the platform must poll agents and this polling generates too much network traffic. NerveCenter helps alleviate this problem by enabling you to determine the interval at which a poll is sent and to turn a poll off. Even more important is NerveCenter's *smart polling* feature. NerveCenter sends a poll to a node only if the poll:

- Is part of a behavior model designed to manage that node
- Can cause a change in the alarm's state

Also, because of NerveCenter's client/server architecture, NerveCenter servers can be distributed so that all polling is done on LANs, and not across a WAN. Furthermore, use of SNMPv2c and v3 features allow SNMP to be utilized both reliably and securely.

Correlating Conditions

Event correlation involves taking a number of detected network conditions and determining:

- How these conditions, or some subset of them, are related
- The underlying cause, or the problem to which these conditions have led

For instance, NerveCenter may look at a large number of events and identify a subset of events that relate to SNMP authentication failures on a managed node. NerveCenter may then determine that the authentication failures were far enough apart that no problem exists, or it may find that several failures occurred within a short period of time, indicating a possible security problem. In the latter case, NerveCenter might notify administrators of the potential problem. In this way, administrators receive one notice about a potential security problem rather than having to browse through a long list of detected conditions and identify the problem themselves.

Detected conditions can be correlated in many ways. In fact, once you start working with NerveCenter, you will help determine how these conditions are correlated yourself. However, there are some typical ways in which NerveCenter finds relationships between conditions.

Several of these methods are discussed in the following sections:

- *Detecting the Persistence of a Condition on page 88*
- *Finding a Set of Conditions on page 89*
- *Looking for a Sequence of Conditions on page 90*

Detecting the Persistence of a Condition

Probably the simplest method of correlating detected conditions is to search for the persistence of a problem. For example, a network administrator might want to know if an SNMP agent sends a link-down trap and that trap is not followed within three minutes by a link-up trap. NerveCenter can track such a link-down condition using a state diagram similar to the one shown in *Figure 8-1*.

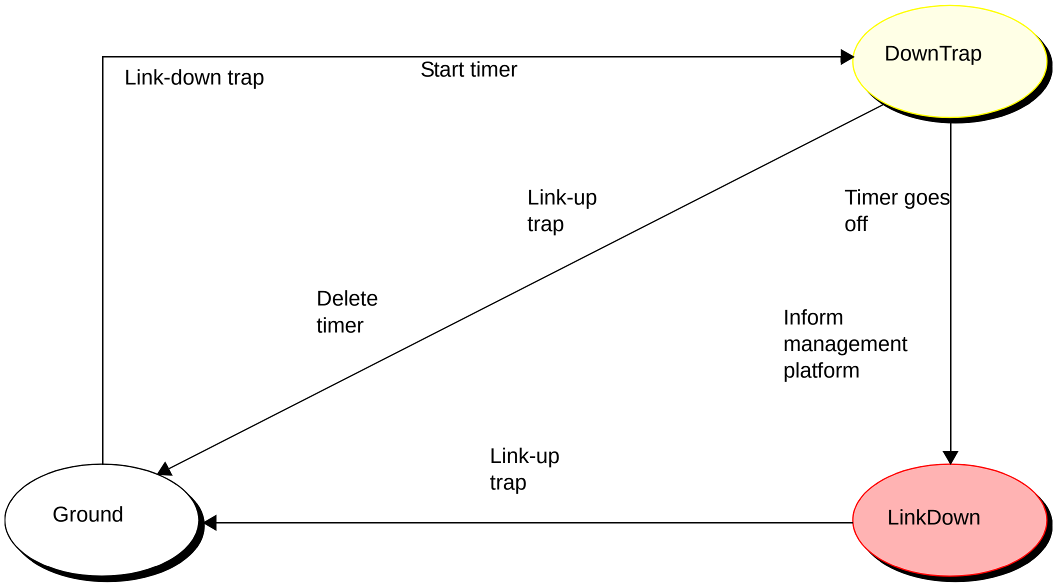


FIGURE 8-1. State Diagram for Detecting a Link-Down Condition

If NerveCenter has this state diagram in memory and is tracking a particular interface for a link-down condition.

- The first time NerveCenter sees a link-down trap concerning that interface, the current state becomes DownTrap, and NerveCenter starts a three-minute timer.
- If NerveCenter receives a link-up trap within three minutes of the link-down trap, the current state reverts to Ground (normal) because NerveCenter is looking for a *persistent* link-down condition. In addition, NerveCenter stops the timer. However, if three minutes expire before a link-up trap arrives, the current state becomes LinkDown, and NerveCenter informs a network management platform that the link is down.
- The current state remains LinkDown until a link-up trap does arrive. At that point, the current state reverts to Ground, and the process begins again.

Finding a Set of Conditions

Another common type of event correlation is identifying a set of conditions. For example, when monitoring the interfaces on a router, you might use the state diagram in *Figure 8-2* to be notified when a low-speed or high-speed interface goes down.

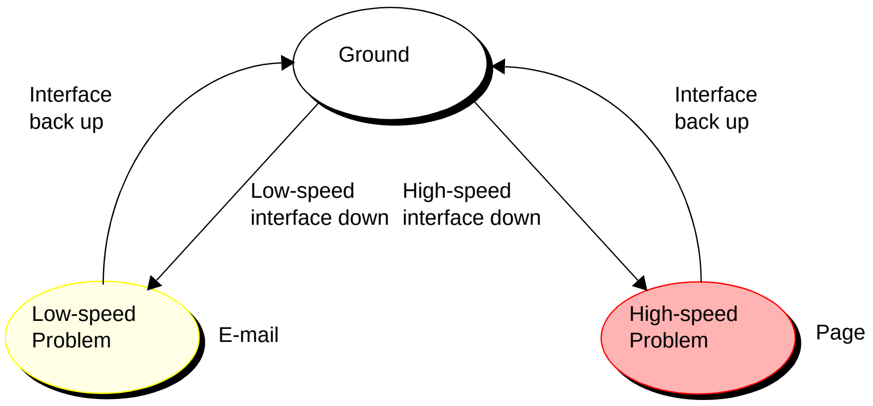


FIGURE 8-2. State Diagram for Detecting a Router Interface Problem

What causes state transitions in this situation? NerveCenter can poll the SNMP agent on the router for the values of the following interface attributes: ifOperStatus, ifAdminStatus, ifSpeed, ifInOctets, and ifOutOctets.

If the poll successfully returns values for these attributes, NerveCenter can then evaluate the expression shown below in pseudocode:

```
if ifOperStatus is down && ifAdminStatus is up &&
(ifInOctets > 0 || ifOutOctets > 0)
      if ifSpeed < 56K
         move to lowSpeedProblem state
      else
         move to highSpeedProblem state
      else
      move to ground state
```

This code is looking for two sets of conditions. The first set is:

- The operational state of the interface is down.
- The administrative status of the interface is up.
- Traffic has been passed on this interface. (If no traffic has been passed, the interface is just coming up.)
- The interface's current bandwidth is less than 56K.

If these conditions are met, a problem exists on an interface that is likely used for dial-up connections. The second set of conditions is the same except for the last condition, which checks whether the interface's current bandwidth is greater than or equal to 56K. If this second set of conditions is met, a problem exists on a higher speed interface. If neither of these sets of conditions is met, the current state should return to, or remain at, Ground.

NerveCenter may detect many conditions concerning an interface before it finds the set of conditions it is looking for. The administrator need not see information about each of these conditions. He or she will be emailed or paged if the interface goes down.

Looking for a Sequence of Conditions

NerveCenter also enables correlation by looking for sequences of conditions. This type of correlation is possible because, each state in a state diagram can look for a different set of conditions. For instance, let's look at a state diagram that NerveCenter uses to track the status of a node and its SNMP agent. The diagram (*Figure 8-3*) includes states for the following conditions:

- The node and its SNMP agent are up.
- The node is up, but its agent is down.
- The node is unreachable.
- The node is down.

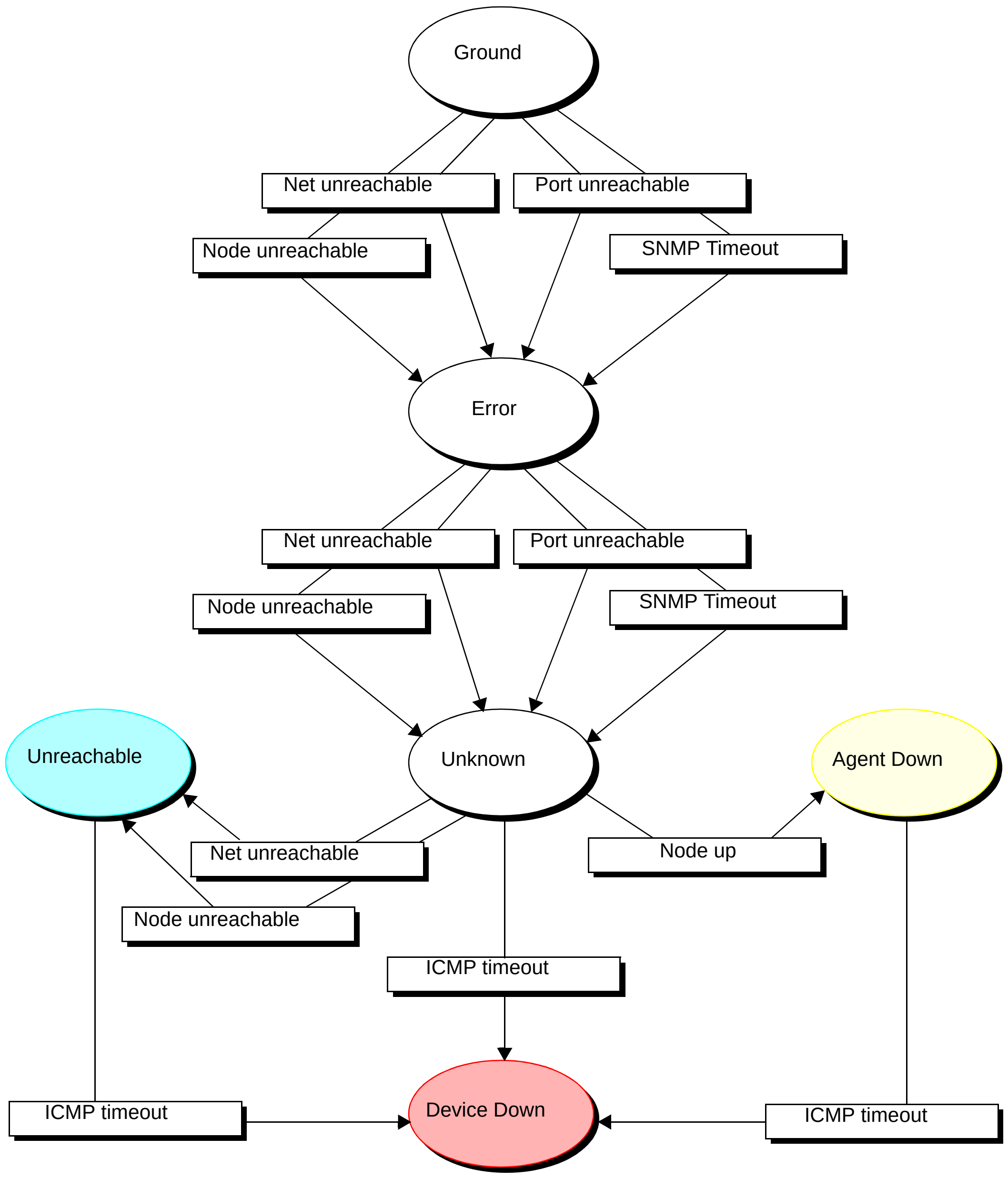


FIGURE 8-3. State Diagram for Determining Node Status

NOTE
A more realistic state diagram for tracking the status of a node would include transitions from the terminal problem states back to Ground.

When checking the status of a node and its SNMP agent, NerveCenter begins by polling the node to see if the node's SNMP agent will return the value of the MIB attribute sysObjectID. If the agent returns this value, the current state remains Ground. However, NerveCenter makes Error the current state if:

- The node, or the network the node is on, is unreachable
- The node is reachable, but the SNMP agent doesn't respond

Similarly, NerveCenter changes the current state to Unknown if it detects for a second time that the node is unreachable or the node's SNMP agent isn't responding.

Once the current state becomes Unknown, though, NerveCenter begins looking for a different set of conditions. NerveCenter checks to see whether the node will respond to an ICMP ping. If it will, NerveCenter knows that the node is up, but its SNMP agent is down. If it receives another network- or node-unreachable message, NerveCenter knows that the node is unreachable. And if the ping times out, NerveCenter knows that the node is down.

This ability of different states to monitor different conditions gives you the ability to correlate *sequences* of conditions. That is, a sequence of two SNMP timeouts followed by a Node up indicates that the node is up but its agent is down. And a sequence of two Node unreachables followed by an ICMP timeout indicates that the node is down.

Responding to Conditions

NerveCenter not only enables you to detect network and system problems, but is able to respond automatically to the conditions it detects. To set up these automated responses, you associate *actions* with state transitions.

The possible actions you can define are discussed in the following sections:

- *Notification on page 93*
- *Logging on page 93*
- *Causing State Transitions on page 94*
- *Corrective Actions on page 94*
- *Action Router on page 95*

Notification

If a particular network or system condition requires the attention of an administrator, the best action to take in response to that condition is to notify the appropriate person. NerveCenter lets you notify administrators of events in the following ways:

- You can send an audible alarm (a beep) to workstations running the NerveCenter Client.
- You can send email to an administrator using either a Microsoft Exchange Server client or SMTP mail.
- You can page an administrator.
- You can send information about a network or system condition to another NerveCenter server. This capability is useful if you have a number of NerveCenter servers at different sites and want these servers to forward information about important events to a central server.
- You can send information about a network or system condition to a network management platform such as IBM Tivoli's Netcool/OMNIbus or Hewlett Packard's OpenView Network Node Manager. Administrators can then be notified of a problem found by NerveCenter using the other management tool's console.

  For more information on integrating NerveCenter with other network management products, see the section *Role in Network Management Strategy on page 29*.

Logging

If you want to keep a record of an event that takes place on your network, you must explicitly log information about the event at the time it occurs. NerveCenter provides three actions that provide for such logging:

- Log to File
- Log to Database (Windows only)
- EventLog

Log to File writes information about an event to a file. Log to Database writes information about an event to the NerveCenter database. The EventLog action writes information about an event to an event or system log.

When you assign a logging action to a behavior model, you have the choice of logging default data or customizing what data you deem relevant. This saves disk space and streamlines information used later for analysis and reporting.

Causing State Transitions

In some behavior models, one alarm needs to cause a transition in another. The action that enables such communication between alarms is called Fire Trigger. This action creates a NerveCenter object called a trigger that can cause a state transition in the alarm from which it was fired or in another alarm.

The Fire Trigger action also lets you specify a delay, so you can request that a trigger be fired in one minute or five hours. This feature is especially useful when you're looking for the persistence of a condition. Let's say that you want to look for three intervals of high traffic on an interface within a two-minute period. When your poll detects the first instance of high traffic, and your alarm moves out of the Ground state, you can fire a trigger with a two-minute delay that will return your alarm to the Ground state—unless a second and third instance of high traffic are detected.

If a third instance of high traffic is detected, you should cancel the trigger you fired on a delayed basis. You do this by adding the Clear Trigger action to the transition from the second high-traffic state to the third.

NerveCenter also includes a Send Trap action. You define the trap to be sent, including the variable bindings, and associate the action with a state transition. When the transition occurs, the trap is sent. The trap can be caught by a NerveCenter trap mask—in which case you can use Send Trap somewhat like Fire Trigger, to generate a trigger—or by any application with SNMP traps.

Corrective Actions

There are a number of NerveCenter actions that you can use to take corrective actions when a particular state transition occurs. These are:

- Command
- Perl Subroutine
- Set Attribute
- Delete Node
- SNMP Set

The Command action enables you to run a script or executable when a particular transition occurs.

The Perl Subroutine action enables you to execute a Perl script as a state-transition action. You first define a collection of Perl scripts and store them in the NerveCenter database; then, you choose one of your stored scripts for execution during a state transition.

The Set Attribute action enables you to set selected attributes of the NerveCenter objects used to build behavior models.

The Delete Node action deletes the node associated with the current state machine from the NerveCenter database. This action is useful if you use a behavior model to determine which nodes you want to monitor and manage.

The SNMP Set alarm action changes the value of a MIB attribute when an alarm transition occurs.

Action Router

The Action Router enables you to specify actions that should be performed when a state transition occurs *and other conditions are met*. To set up these conditional actions, you add the Action Router action to your state transition. Then, you use the Action Router tool to define rules and their associated actions.

For example, let’s assume that you want to be notified about a state transition only if the transition puts the alarm in a critical state. You can define the following rule:

```
$DestStateSev eq ‘Critical’
```

Then define the action you want taken if the severity of the destination state is Critical, for example, a page. You will be paged if:

- The Action Router action is associated with the current state transition
- The destination state for the transition is Critical

Action Router rules can be constructed using many variables that NerveCenter maintains; for instance, you can also construct rules based on:

- The name of the alarm
- The day of the week
- The time of day
- The name or IP address or group property of the node being monitored
- The name of the trigger that caused the state transition
- The name of the alarm’s property
- The name or severity of the origin state
- The contents of a trap
- The contents of the varbind data associated with a trap or a poll

The NerveCenter Node List

To detect and correlate network events, NerveCenter must have basic information about each managed node it monitors. NerveCenter obtains this data through several means.

NerveCenter can obtain node information from any or all of the following sources:

- A network management platform, such as HP OpenView Network Node Manager
- The NerveCenter Discovery behavior model
- An administrator’s manual entries

NOTE

Though NerveCenter supports SNMPv1, v2c, and v3, when NerveCenter obtains nodes from a platform, the platform does not provide SNMP version information. By default, NerveCenter deems the SNMP agents on these nodes to be SNMPv1.

If you want NerveCenter to attempt SNMP version classification automatically for the nodes it receives from your platform, you must enable auto-classification. Then, NerveCenter can classify the correct SNMP version for each node with each resynchronization. Refer to *Managing SNMP Settings on page 125* for more information about SNMP auto-classification.

NOTE

To manage SNMPv3 nodes, you must use NerveCenter as your trap source regardless of the node source you configure. Refer to *Changing the SNMP Trap Source on page 55* for more information about the SNMP trap source.

When NerveCenter uses information obtained by a network management platform, it does not use the platform’s database as its repository for managed nodes. Instead, it stores node information in its own database in a node list.

There are a few reasons for NerveCenter maintaining a node list in its own database:

- There may be a considerable distance between the platform’s database and NerveCenter, making frequent access time-consuming and costly.
- NerveCenter adds configuration data to the node data that the management platform does not necessarily provided.
- Administrators have the option of adding nodes not in the platform’s node database to the node list in NerveCenter’s database.

Filtering Nodes

Before populating the node list in NerveCenter's database, it is important to determine which nodes NerveCenter will manage, as NerveCenter does not need to monitor every node on your network. There are a few methods for restricting which nodes will be placed in NerveCenter's node list:

- *Filtering Using Node Capabilities on page 97*
- *Filtering Using a Node System Object Identifier on page 98*
- *Filtering Nodes by IP Address on page 99*
- *Filtering Nodes by Hostnames on page 104*

Filtering Using Node Capabilities

You can monitor nodes with particular capabilities, which your network management system typically assigns to a node to determine the applicable activities such as isRouter, isHub, and isIP.

NOTE
Filtering by capabilities is available only when your network management platform has assigned specific capabilities to a node.

TO FILTER USING A NODE'S CAPABILITIES

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server.

   See the section *Connecting Administrator to a NerveCenter Server on page 43*.

2. Select the **Node Source** tab.
3. In the **Wanted Capabilities** field, type the name of the specific capability desired.

   If this field is left blank, NerveCenter will not filter using a node's capability.

4. To enter multiple capabilities, separate each with a space.

   NerveCenter will monitor any node that matches at least *one* of the capabilities in the list.

5. Select **Save**.

   The NerveCenter Server adds the new capabilities filter. It also closes and opens a new connection with the platform adapter. NerveCenter automatically resynchronizes with your network management platform database. New nodes will be added. Any node that is marked Autodelete (the default) will be deleted.

Filtering Using a Node System Object Identifier

NerveCenter allows you to monitor managed nodes according to their particular system object identifiers (OIDs). A node's System Object ID is an SNMP MIB-II object in the system group. It identifies the SNMP agent software running on the device. It is, however, commonly used to identify the type and vendor of the device because a particular vendor's agent usually runs on that vendor's devices.

TO FILTER USING A NODE'S SYSTEM OBJECT IDENTIFIER

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server.

   See *Connecting Administrator to a NerveCenter Server on page 43*.
2. Select the **Node Source** tab.
3. In the **System Object Ids** field, type the name of the system OID wanted.

   If this field is left blank, NerveCenter will not filter using a node's system OID.
4. To enter multiple OIDs, separate each with a space.

   NerveCenter will monitor any node that matches at least *one* of the OID in the list.

   For example, an administrator may want to restrict NerveCenter to nodes running SNMP agents from either Cisco or Hewlett-Packard nodes by typing the following:

   ```
   1.3.6.1.4.1.9 1.3.6.1.4.1.11
   ```

   Any device with an OID matching either of these numbers will be included in the NerveCenter nodes database.
5. Select **Save**.

   The NerveCenter Server adds the new OID filter. It also closes and opens a new connection with the platform adapter. NerveCenter automatically resynchronizes with your network management platform's database. New nodes will be added. Any node that is marked Autodelete (the default) will be deleted.

Filtering Nodes by IP Address

In addition to filtering nodes by OIDs and capabilities, NerveCenter allows you to filter out all nodes that do not belong to one or more subnets. NerveCenter determines the subnet by combining a specific IP address with a subnet mask. NerveCenter can filter by subnets of both Class B and Class C networks. In Class B networks, the first two octets specify the network while in Class C networks the first three octets identify the network.

Table 8-1 and *Table 8-2* illustrate some filter configurations and their results:

TABLE 8-1. Sample Subnet Filters and Their Results for a Class C Network

IP address	Subnet mask	Result
134.204.179.0	255.255.255.0	All nodes on subnet 134.204.179.0 are included. For example, 134.204.179.7 is included.
197.22.44.0	255.255.255.240	All nodes 1-15 on subnet 197.22.44.0 are included. For example, 197.22.44.5 is included but 197.22.44.35 is excluded.
134.204.179.0 197.22.44.0	255.255.255.0 255.255.255.240	All nodes on subnets 134.204.179.0 and nodes 1-15 on 197.22.44.0 are included. For example, both 134.204.179.7 and 197.22.44.5 are included.

TABLE 8-2. Sample Subnet Filters and Their Results for a Class B Network

IP address	Subnet mask	Result
132.45.0.0	255.255.0.0	All nodes on subnet 132.45.0.0 are included. For example, 132.45.174.7 is included.
132.45.0.0	255.255.240.0	Nodes 1.0 - 15.255 are included. For example 132.45.14.231 is included but 132.45.174.7 is excluded.

In addition to filtering out all but an entire subnet, NerveCenter allows you to exclude a specific node or range of nodes within the remaining subnet.

Table 8-3 illustrates some filter configurations with exclusions and their results:

TABLE 8-3. Sample Subnet Filters with Exclusions and Their Results

IP address	Subnet mask	Excluded node(s)	Result
134.204.179.0	255.255.255.0	40	All nodes on subnet 134.204.179.0 except node 40 are include. For example, 134.204.179.7 is included but 134.204.179.40 is excluded.
134.204.179.0	255.255.255.0	40-55	All nodes on subnet 134.204.179.0 except nodes 40-55 are include. For example, 134.204.179.7 is included but 134.204.179.40 and 134.204.179.52 are excluded.
132.45.0.0	255.255.0.0	63.5	All nodes on subnet 132.45.0.0 except node 63.5 are included.

NOTE
You can filter nodes that have been discovered by NerveCenter, provided by the platform node source, or imported from a node file.

NerveCenter can automatically or manually determine the subnet criteria used to filter nodes by IP address.

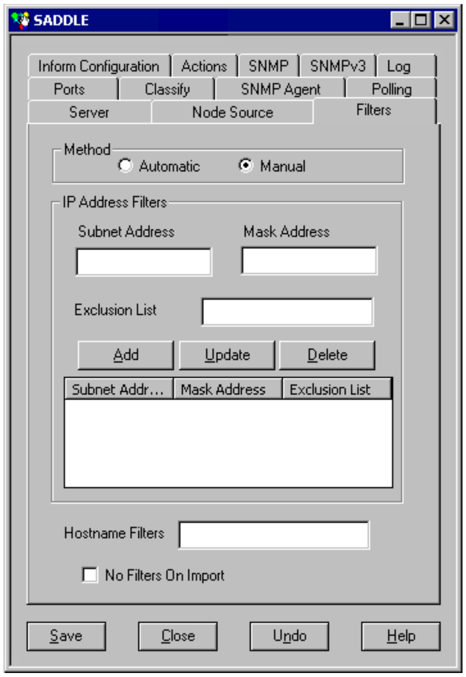
TO CONFIGURE NERVECENTER TO DETERMINE SUBNET CRITERIA AUTOMATICALLY

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server.

   See *Connecting Administrator to a NerveCenter Server on page 43*.

2. Select the **Filters** tab.

   The Filters tab is displayed.



Method: Automatic [✓ Manual]

IP Address Filters

Subnet Address:

Mask Address:

Exclusion List:

Subnet Addr...	Mask Address	Exclusion List

Hostname Filters:

☐ No Filters On Import

3. In the Method area, select **Automatic**.

   Setting the method to automatic tells NerveCenter to ignore any address filters you enter and use instead the server's masks as a filter. NerveCenter calculates the subnet address and mask using the IP address and mask of each network interface card on the server.

4. Select **Save**.

   NerveCenter will now automatically use the server's masks as a filter.

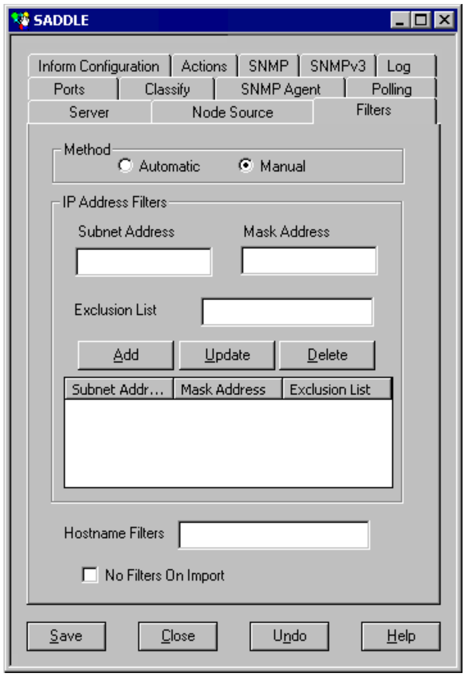
To set the subnet criteria manually

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server.

   See *Connecting Administrator to a NerveCenter Server on page 43*.

2. Select the **Filters** tab.

   The Filters tab is displayed.



Method: Automatic [✓ Manual]

IP Address Filters

No Filters On Import [ ]

3. In the Method area, select **Manual**.

   When the method is set to manual, NerveCenter will only use the subnet addresses listed in IP Address Filters area. If the IP Address Filters list is empty, NerveCenter ignores a node's subnet when determining if the node will be part of the node database.

4. In the **Subnet Address** field, type the appropriate subnet. In the **Mask Address** field, type the appropriate subnet mask.

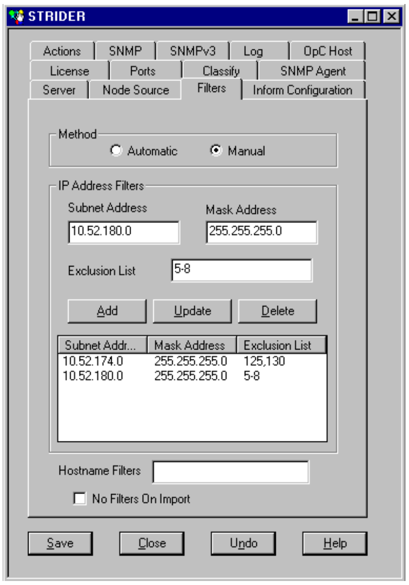
   A node's subnet address combines the node's IP address with the subnet mask.

5. In the Exclusion List field, enter all the nodes you want excluded from the subnet address. To exclude more than one node, separate each number with a comma without a space. To exclude a continuous range of nodes, use a hyphen to separate the minimum and maximum number by a hyphen.

   CAUTION
   Once you have added a node using the IPSweep behavior model, you cannot use an IP exclusion to delete it from the database. Once the node is in the database, the IP exclusion filter is not applied to it. The IP exclusion applies only to new nodes discovered after the filter is established.

6. Select **Add**.

   The subnet address and mask address will be added to the IP Address Filters list.



Method: Automatic [✓ Manual]

IP Address Filters

Subnet Address: 10.52.180.0

Mask Address: 255.255.255.0

Exclusion List: 5-8

Subnet Addr...	Mask Address	Exclusion List
10.52.174.0	255.255.255.0	125,130
10.52.180.0	255.255.255.0	5-8

Hostname Filters:

☐ No Filters On Import

7. Select **Save**.

8. To filter by additional IP addresses and masks, repeat steps 4 and 7.

NerveCenter monitors any address falling within the subnet and not excluded by the filter.

Filtering Nodes by Hostnames

In addition to filtering nodes by OIDs, capabilities, and IP Addresses, NerveCenter allows you to filter out nodes by hostname. Filtering by hostname works in conjunction with IP Filters. If a node matches on either an IP filter or a hostname filter, the node will be filtered.

The hostname filter restricts imported node lists and node source from OVPA. Nodes discovered by SNMP traps are not affected by hostname filters because discovered nodes do not always have hostname information.

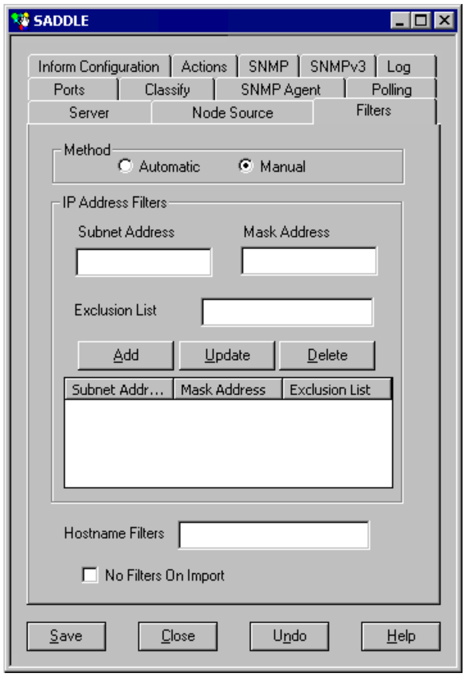
TO CONFIGURE HOSTNAME FILTERS

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server.

   See *Connecting Administrator to a NerveCenter Server on page 43*.

2. Select the **Filters** tab.

   The Filters tab is displayed.



Method: ( ) Automatic [✓ Manual]

3. In the Hostname Filters field, enter the hostnames you want to exclude, separating names with a space.

   You can use an asterisk (*) as a wild card. For example, ***server** filters out all hostnames ending with server and ***router*** filters out all hostnames containing the word router.

   NOTE
   There is no guarantee that OVPA or node lists will always provide fully qualified host names. Therefore, filtering by domain names may not work if the node list does not contain fully qualified hostnames.

4. Select **Save**.

   NerveCenter now filters by hostnames.

Enabling and Disabling IP and Hostname Filters

You can define IP Address and hostname filters to limit what nodes you add to your NerveCenter node list. This is important when using discovery methods such as IPSweep, so that you do not try to include the entire internet. However, if you have a node list you want to import of known hosts you want to monitor, you may not want to apply the IP and Hostname filters.

IP filters apply to the following:

- Importing node lists
- Populating the node lists with a network management platform
- Discovering nodes from traps, if you select IP Filter from the Discover Nodes from Traps list on the Server tab.

Hostname filters apply to the following:

- Importing node lists
- Populating the node lists with a network management platform

You can disable IP and hostname filters if you do not want to use them for an import from a node list or management platform.

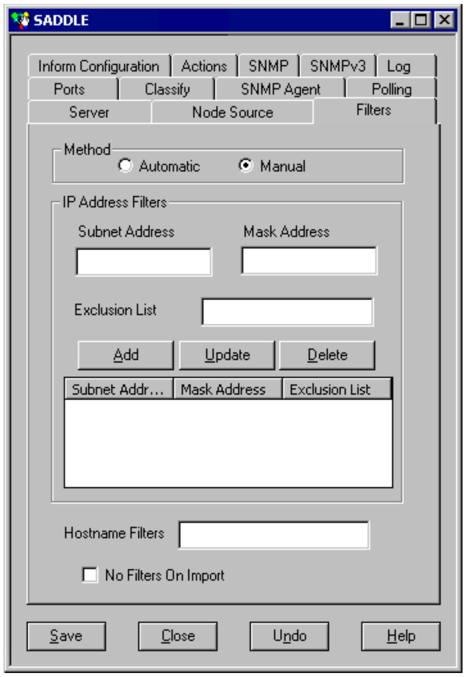
To enable or disable IP and hostname filters

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server.

   See *Connecting Administrator to a NerveCenter Server on page 43*.

2. Select the **Filters** tab.

   The Filters tab is displayed.



Method: ( ) Automatic [✓ Manual]

IP Address Filters

Subnet Address:

Mask Address:

Exclusion List:

Subnet Addr...	Mask Address	Exclusion List

Hostname Filters:

☐ No Filters On Import

3. Select **No Filters On Import** to disable IP and hostname filters; clear it to enable IP and hostname filters.

4. Click **Save**.

Processing Traps from Unknown Nodes

NerveCenter offers a lot of flexibility in filtering the managed nodes it monitors. By setting node filters, you are telling NerveCenter to ignore any traps sent by nodes not stored in the node list. This keeps the node list to a size that is manageable and acceptable under your license agreement.

However, there may be times you will want a NerveCenter Server to process a trap from a node not found in its node list, but in the node list of another NerveCenter Server. For example, in the following diagram, a NerveCenter Server responsible for monitoring a LAN (Server A) needs to pass a trap along to a WAN-level NerveCenter Server (Server B).

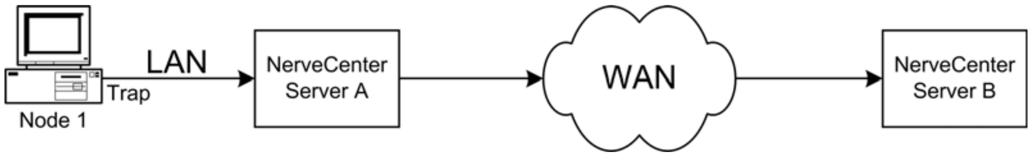


FIGURE 8-4. An Example of a Trap Being Processed from an Unknown Node

In this case, Node 1 is in the node list of Server A. Node 1 sends a trap to Server A, which in turn is passed along to Server B. Because Node 1 is being managed by Server A, it will not appear in Server B's node list. Server B needs to process the trap from Node 1. Server B could be set up to add unknown nodes to its node list whenever it receives a trap, but then both Server A and Server B will be monitoring Node 1. How can Server B process the trap from Node 1 without first adding the node to its node list?

The Process traps from unknown nodes feature allows a NerveCenter Server receiving a trap from an unknown node to process that trap if the trap is associated with an Enterprise scope alarm.

NOTE

The Process traps from unknown nodes feature will only process traps associated with Enterprise scope alarms.

At the same time, the Process traps from unknown nodes feature will keep the NerveCenter Server from adding the node to its node list. This ensures that the NerveCenter Server does not exceed the node limit allowed by its license. It also avoids a situation in which two NerveCenter Servers are responsible for the same node.

NOTE

To learn how a NerveCenter Server adds an unknown node to its node list when receiving a trap from that node, see *Adding Nodes Discovered from Traps on page 117*.

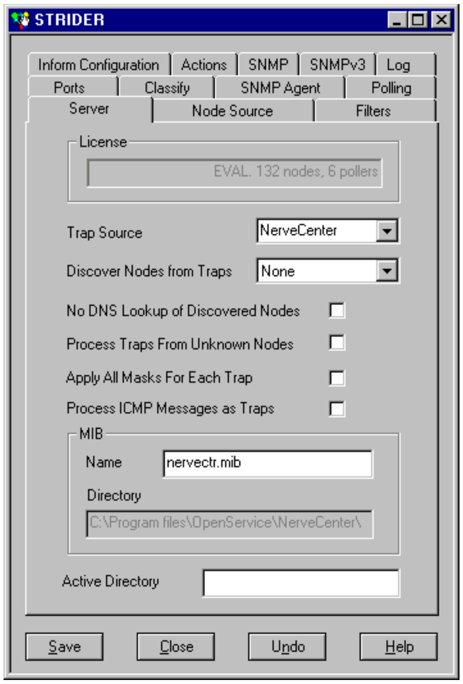
TO PROCESS TRAPS FROM UNKNOWN NODES

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server.

   See *Connecting Administrator to a NerveCenter Server on page 43*.

2. Select the **Server** tab.

   NerveCenter displays the Server tab.



License: EVAL. 132 nodes, 6 pollers
Trap Source: NerveCenter
Discover Nodes from Traps: None
No DNS Lookup of Discovered Nodes ☐
Process Traps From Unknown Nodes ☐
Apply All Masks For Each Trap ☐
Process ICMP Messages as Traps ☐
MIB
Name: nervectr.mib
Directory: C:\Program files\OpenService\NerveCenter\
Active Directory:

3. Select the **Process Traps From Unknown Nodes** box.

   If this box is not selected, NerveCenter ignores any traps from a node not in its node list.

4. Select **Save**.

The NerveCenter Server will now process traps associated with Enterprise scope alarms even when those traps are received from nodes not appearing in the server’s node list.

You can also use a Perl function, AddNode(), to add nodes to the NerveCenter database. See *AddNode() Function in Designing and Managing Behavior Models* for details.

Initially Populating the Node List

To detect and correlate network events, NerveCenter must have basic information about each of the managed nodes it monitors. Once you have configured the proper node filters (see *Filtering Nodes on page 97*), you must tell NerveCenter where to look for the initial data it will use to populate its node list.

Though you can manually add nodes to NerveCenter, it is easy to configure NerveCenter to discover nodes (*Figure 8-5*). When NerveCenter discovery is enabled, if the database does not already contain a node that sends it an SNMP trap or NerveCenter inform, NerveCenter adds that node to the database.

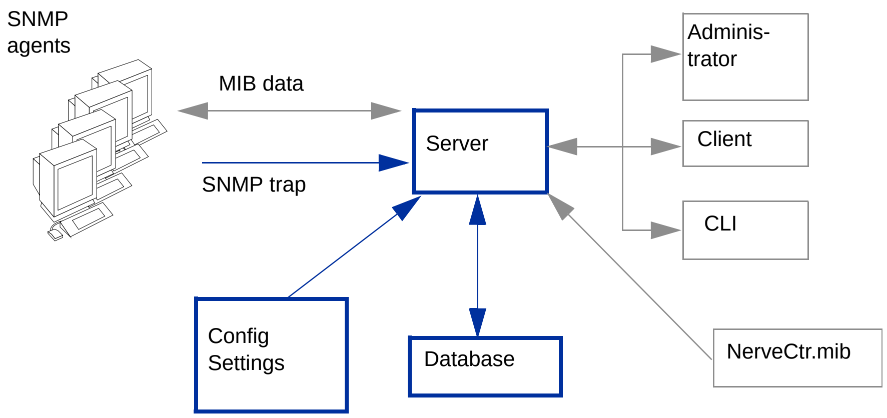


FIGURE 8-5. NerveCenter Configured To Discover Nodes

After installing NerveCenter, you can use the NerveCenter Administrator to define subnet IP filters that limit the sets of nodes NerveCenter can monitor. These filter values are stored with the NerveCenter configuration settings.

When a trap is received from a node, NerveCenter compares the node against those that are already in its database and confirms whether the node falls within the subnet IP filters you defined. Nodes that fall within the subnet range but are not in the database are added to the database.

NOTE

NerveCenter can be configured to process SNMP traps from nodes residing outside the defined subnet filters. These nodes, however, are not added to the database. See *Filtering Nodes by IP Address on page 99* for more details.

8 Managing Node Data

There are three main methods for populating NerveCenter’s node list:

- *Populating Using a Network Management Platform as a Data Source on page 110*
- *Populating Using the IPSweep Behavior Model on page 112*
- *Populating the Node List Manually on page 114*

Populating Using a Network Management Platform as a Data Source

NerveCenter is able to receive information about some or all of the nodes managed by your network management platform. NerveCenter is able to retrieve data about managed nodes from the following HP OpenView Network Node Manager.

NOTE

To use your network management platform to populate NerveCenter’s node database, you must have the NerveCenter OpenView Platform Adapter (OVPA) installed and running.

To populate NerveCenter’s node list using your network management platform, you must specify it as a source for the node data. Each NerveCenter database populates its node list from just one network management platform database. Depending on your filtering, the database may contain all the nodes or just a subset. In either case, there is just one source of the information.

CAUTION

If you wish to map system Object Identifiers (OID) to NerveCenter property groups, you must make the necessary configurations in the NerveCenter Client before naming the node data source. (See *Using OID to Property Group Mappings in Designing and Managing Behavior Models*.) After NerveCenter initially populates its node list, any subsequent mapping of OIDs to property groups affect only new nodes added to the node list.

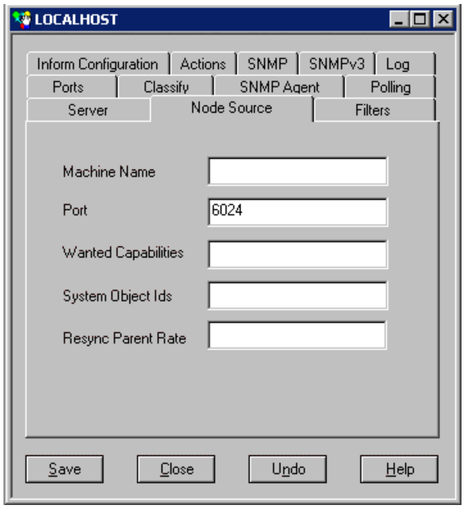
TO USE YOUR NETWORK MANAGEMENT PLATFORM AS A NODE DATA SOURCE

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server.

   See *Connecting Administrator to a NerveCenter Server on page 43*.

2. Select the **Node Source** tab.

   NerveCenter displays the Node Source tab.



Machine Name
Port 6024
Wanted Capabilities
System Object Ids
Resync Parent Rate

3. In the **Machine Name** field, type the name or IP address of a host that runs your network management platform.

   If the Machine Name field is left blank, NerveCenter does not retrieve nodes.

4. In the **Port** field, type the number of the port used to communicate with the platform adapter process on the host. The default is 6024.

   The platform adapter must be configured to listen on the same port specified in this field.

5. Select **Save**.

NerveCenter will now retrieve its initial node data from your network management platform's database. IP and hostname filters can apply to what node data is imported from the network management platform. See *Enabling and Disabling IP and Hostname Filters on page 105* for details about using filters with node lists.

Populating Using the IPSweep Behavior Model

There may be situations in which you will not want to use a network management platform to populate NerveCenter's node list initially. Such situations could include:

- Your network does not contain an applicable network management platform or you are using NerveCenter as a stand-alone application.
- A large distance or a costly link separates the platform from the nodes that NerveCenter will be monitoring locally.

The IPSweep behavior model is intended to handle these situations. IPSweep allows NerveCenter to detect unknown nodes and add them to the node database, provided they fall within the set filters. *Figure 8-6* shows how IPSweep fits in with the other NerveCenter components.

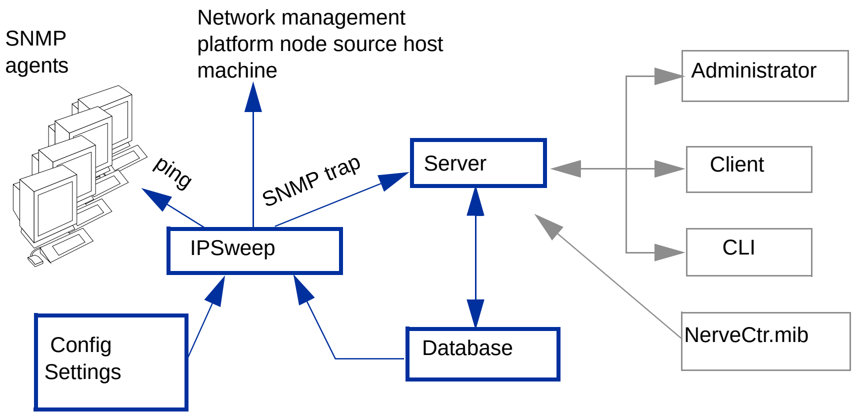


FIGURE 8-6. NerveCenter Configured to Discover Nodes

IPSweep extracts subnet IP filter information from the NerveCenter configuration settings, obtains node information from the NerveCenter database, identifies nodes that fall within the subnet range but are not in the database, and sends those nodes a ping (ICMP echo request). If the ping returns a response, IPSweep issues an SNMP trap either to a host specified as the node source or, if no node source is specified, to the local NerveCenter. In a standalone configuration, the trap is sent to NerveCenter. Once NerveCenter receives the trap, the node is added to the NerveCenter database.

IPSweep does not populate the NerveCenter server database; it sends a trap if it gets a ping response from an IP address, at which point the server can add the unknown node to its database.

NOTE

If you are using NerveCenter in Windows with a Domain Name Server (DNS), the IPSweep behavior model requires that **Enable DNS for Windows Resolution** be selected on the Protocols > TCP/IP > WINS tab of the Network Control Panel.

TO POPULATE THE NODE LIST USING THE IPSWEEP BEHAVIOR MODEL

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server.

   See *Connecting Administrator to a NerveCenter Server on page 43*.

2. Select the **Filters** tab and configure NerveCenter to filter by the appropriate subnet criteria. See *Filtering Nodes by IP Address on page 99*.

   CAUTION

   If the IP Filter box is empty, the IPSweep behavior model's primary application ipsweep will not run. This precaution is to prevent NerveCenter from trying to discover all the nodes on the Internet.

3. Select the **Server** tab.

   NerveCenter displays the Server tab.

4. Select from the **Discover Nodes from Traps** list **All** or **IP Filter**.

   NOTE

   IP Filter only functions if IP Filters are enabled as described in *Enabling and Disabling IP and Hostname Filters on page 105*. If the **Discover Nodes from Traps** field is set to **None**, the IPSweep behavior model will not work.

5. (Optional) Select **No DNS Lookup of Discovered Nodes** to improve performance.

   By default, No DNS Lookup of Discovered Nodes is not selected and NerveCenter performs a DNS lookup for each unknown node. If you select this box, the nodes are added by IP address and NerveCenter does not attempt a DNS lookup.

6. Select **Process Traps From Unknown Nodes**.

7. If you want NerveCenter to turn on the IPSweep behavior model every time the NerveCenter Server is started, select **Enable Discovery at Startup**.

8. From NerveCenter Client, turn on the IPSweep behavior model. See *Enabling the IPSweep Alarm in Designing and Managing Behavior Models* for details.

NerveCenter periodically runs the script ipsweep. As NerveCenter discovers unknown nodes that fall within its IP filters it will add them to the node list. For more information about NerveCenter's IPSweep behavior model, see *Using IPSweep Behavior Model in Designing and Managing Behavior Models*.

Populating the Node List Manually

The two most popular methods of initially populating NerveCenter's node list are:

- *Populating Using a Network Management Platform as a Data Source on page 110*
- *Populating Using the IPSweep Behavior Model on page 112*

An alternative is to populate the NerveCenter node list by hand. For each node that NerveCenter will manage, you must use the Node Definition window to specify the node's data, including:

- Name
- Address
- Community string
- Property group

Because defining your own node data demands a considerable amount of attention, this alternative is recommended only for the smallest number of managed nodes.

To populate NerveCenter's node list manually, follow the steps described in *Adding and Deleting Nodes Manually on page 119*.

Maintaining the Node List

To detect and correlate network events, NerveCenter must have basic information about each of the managed nodes it is monitoring.

Once you have initially populated the node list (see *Initially Populating the Node List on page 109*), NerveCenter will need a way to adapt its node list to reflect changes in your network topology. NerveCenter offers several methods for adding and deleting nodes.

These methods include:

- *Synchronization with Your Network Management Platform on page 115*
- *Adding Nodes Discovered from Traps on page 117*
- *Adding and Deleting Nodes Manually on page 119*

Table 8-4 illustrates how each method affects NerveCenter's node list.

TABLE 8-4. Maintaining NerveCenter's Node List

Method	Adds Nodes	Deletes Nodes	Changes Node Data
Resync with the platform	✓	✓	✓
Discover nodes from traps	✓		
IPSweep behavior model	✓		
Manual additions, deletions, and changes	✓	✓	✓

Synchronization with Your Network Management Platform

Over time, a network's topology will change. Eventually your network management platform will add newly discovered devices to its database. It will also delete nodes and change node information. If NerveCenter depends on your network management platform for the data in its node list, it will need to adapt to reflect these changes.

NerveCenter will automatically update its node list to keep in sync with your network management platform's node data. This occurs in the following situations:

- When your network management platform adds a node to its node database. After NerveCenter verifies the node meets the criteria set by its filters, it will add the node to its node list.

- When your network management platform deletes a node from its node database. NerveCenter will delete from its node list any node that is set to Autodelete. Autodelete is the default setting for any new node added to the node list. This setting can be changed in the node's Node Definition Window in the NerveCenter Client. (See *Discovering and Defining Nodes in Designing and Managing Behavior Models*.)
- When your network management platform changes information about a node in its node database. NerveCenter will make any necessary changes to its node data, including changes in the community string, address, parenting information or the managed/unmanaged state.

NOTE

If your network management platform unmanages a node in the NerveCenter node list, the unmanaged state will be updated in NerveCenter. However, if your network management platform unmanages a node not found in NerveCenter's node list, the node will not be added to NerveCenter.

Most often, the node list will only be updated a node at a time. Occasionally, NerveCenter will need to perform a complete resynchronization with the platform. A resynchronization gathers from the platform the most current node data for all nodes. This occurs in the following situations:

- The NerveCenter Server is started and successfully connects to the OpenView Platform Adapter (OVPA).
- A connection between the NerveCenter Server and the node source successfully reconnects after being broken.
- The NerveCenter administrator changes the way in which NerveCenter filters by capabilities or system Object Identifiers (OIDs).
- A user manually chooses **Resync** in the **Server** menu of the NerveCenter Client.

The **Machine Name** field on the **Node Source** tab of the NerveCenter Administrator specifies the name of the host running the platform resynchronizing with NerveCenter. (See *Populating Using a Network Management Platform as a Data Source on page 110* for more details on how to declare a node data source.)

The **Node Source** and **Filters** tabs also specify the parameters NerveCenter uses to filter node data. (See *Filtering Nodes on page 97*.)

Anyone administering NerveCenter should be aware of two important scenarios involving changes to your network management platform's database:

- If the name changes in your network management platform's database, NerveCenter considers it to be a new node.
- If a node is unmanaged in one of your network management platform maps but is managed in another, the node remains in the managed state in NerveCenter's node data.

CAUTION

Since your network management platform's node is matched to a NerveCenter node using its name, you should use care when changing NerveCenter's node configurations. Resynchronization adds nodes when it cannot find names that match your network management platform's map information. Therefore, if you change a node's name in the Node Definition window, resynchronization will not find a match and will add a node, resulting in two nodes with the same address but different names.

Adding Nodes Discovered from Traps

NerveCenter occasionally receives traps from nodes not included in its node list. In this situation, NerveCenter must make two decisions:

- Whether to process the trap or ignore it (as described in *Processing Traps from Unknown Nodes on page 107*).
- Whether to add the unknown node to its node list. This section explains how to configure this setting.

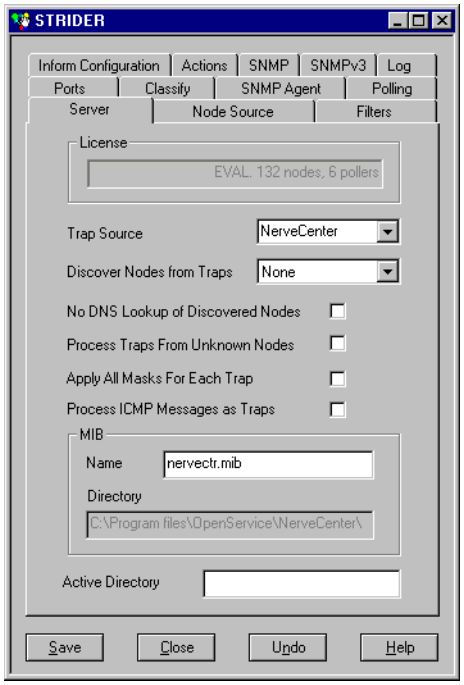
TO CONFIGURE THE WAY UNKNOWN NODES AFFECT NERVECENTER'S NODE LIST

1. Open NerveCenter Administrator and connect to the appropriate Server.

   See *Connecting Administrator to a NerveCenter Server on page 43*.

2. Select the **Server** tab.

   NerveCenter displays the Server tab.



License: EVAL. 132 nodes, 6 pollers

Trap Source: NerveCenter

Discover Nodes from Traps: None

No DNS Lookup of Discovered Nodes ☐

Process Traps From Unknown Nodes ☐

Apply All Masks For Each Trap ☐

Process ICMP Messages as Traps ☐

MIB

Name: nervectr.mib

Directory: C:\Program files\OpenService\NerveCenter\

Active Directory

3. In the **Discover Nodes from Traps** list, select nodes to place in the NerveCenter node list:

   - **None** - NerveCenter will not add any unknown nodes to the node list.
   - **IP Filter** - NerveCenter will add to its node list an unknown node sending a trap *only* if it meets the criteria established by the node filters.

     NOTE
     IP Filter only functions if IP Filters are enabled. See *Enabling and Disabling IP and Hostname Filters on page 105* for details.

   - With an **All** setting, NerveCenter adds to its node list any unknown node sending a trap, regardless of the node filters.

4. (Optional) Select **No DNS Lookup of Discovered Nodes** to improve performance.

   By default, No DNS Lookup of Discovered Nodes is not selected and NerveCenter performs a DNS lookup for each unknown node. If you select this box, the nodes are added by IP address and NerveCenter does not attempt a DNS lookup.

5. Select **Save**.

Any time NerveCenter receives a trap from a node not listed in its node list, NerveCenter handles the node according to the criteria you set in the **Discover Nodes from Traps** list box.

NOTE

NerveCenter sets autodeletes any unknown nodes added from traps. If NerveCenter adds an unknown node discovered from a trap but your network management platform fails to add it, the node will be deleted at the next resynchronization.

Adding and Deleting Nodes Manually

Occasionally, you may want to monitor one or more nodes that do not match the parameters set by the node filters. Or you may want to delete a node or a group of nodes. NerveCenter gives you the ability to alter the node list to your own specifications.

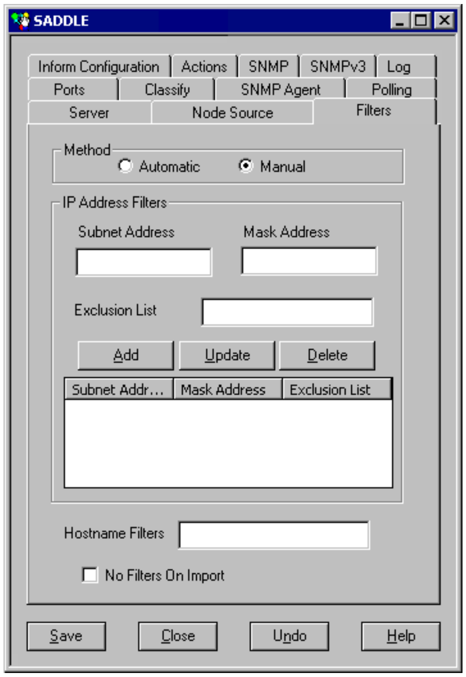
When maintaining the node list manually, you have the following options:

- **Adding a Node Manually** — You add nodes in the NerveCenter Client module, as described in *Defining Nodes Manually in Designing and Managing Behavior Models*.
- **Deleting a Node Manually** — You delete nodes in the NerveCenter Client module, as described in *Deleting Objects in Designing and Managing Behavior Models*.
- **Filter Out a Node That was Manually Deleted** — If you do not complete this step, the nodes you delete manually will be added the next time a resynchronization occurs between NerveCenter and the network management platform's database or the next time the IPSweep behavior model runs the ipsweep script. Following are instructions for filtering out a node.

TO FILTER OUT A NODE THAT WAS MANUALLY DELETED

1. Open the NerveCenter Administrator and connect to the appropriate NerveCenter Server.
2. Select the **Filters** tab.

   The Filters tab is displayed.



3. In the **IP Address Filters** list, select the subnet address of the node you want to delete.
4. In the **Exclusion List** field, enter the nodes you want deleted from the node list. Separate each node with a comma, but no space; use hyphens to exclude a continuous range of nodes.

   NOTE
   If you do not complete this step, deleted nodes will be re-added at the next resynchronization between NerveCenter and the network management platform or the next time the IPSweep model runs the ipsweep script.

5. Select **Update**.
6. Select **Save**.

The nodes are permanently deleted from NerveCenter's database.

Troubleshooting: Managing Node Data

The following list contains some common problems users have when managing node data.

NerveCenter is not filtering a node by a capability

Problem: Filtering by capabilities is available only when a network management platform has assigned a specific capability to a node.

Solution: Have your network management platform assign a capability to the node.

See your network management platform's documentation for details.

Problem: The OVPA command line switch -ignoreCapability has been turned on.

Solution: From the command line, start OVPA without the -ignoreCapability switch.

For more details see *Integrating NerveCenter with a Network Management Platform*.

After setting an IP filter, a node that should be masked out still appears in the node list

Problem: IP filters only exclude additional nodes from being added to a node list. It does not actively delete nodes in the node list.

Solution: Either manually delete the node or force a resynchronization with your network management platform.

See *Adding and Deleting Nodes Manually on page 119* or *Synchronization with Your Network Management Platform on page 115*.

Even though I have enabled Process Traps From Unknown Nodes, NerveCenter does not update its node list when it receives a trap from an unknown node

Problem: Selecting the **Process Traps From Unknown Nodes** box does not affect NerveCenter's node list. It only tells NerveCenter to process traps.

Solution: If you want NerveCenter to update its node list based on traps received from unknown nodes, set Discover Nodes from Traps to Filter or All.

See *Adding Nodes Discovered from Traps on page 117*.

NerveCenter does not recognize my network management platform as a valid source of node data

Problem: Currently NerveCenter is able to retrieve node data from HP OpenView Network Node Manager.

Solution: Use the IPSweep behavior model to obtain node data.

See *Populating Using the IPSweep Behavior Model on page 112*.

NerveCenter is not receiving node data from my network management platform

Problem: They are using different ports.

Solution: Configure the node data source port number to be the same as the platform adapter's port.

See *Populating Using a Network Management Platform as a Data Source on page 110*.

The IPSweep behavior model will not work

Problem: Windows is configured with Domain Name Server.

Solution: Enable the **Enable DNS for Windows Resolution** feature.

See *Populating Using the IPSweep Behavior Model on page 112*.

Problem: There are no IP filters. NerveCenter will not discover nodes unless there are IP filters. This precaution is to prevent NerveCenter from trying to discover all the nodes on the Internet.

Solution: Set the appropriate IP filters.

See *Filtering Nodes by IP Address on page 99*.

Problem: The NerveCenter Server is not set to discover nodes from traps.

Solution: Set **Discover Nodes from Traps** to **All** or **Filter**.

See: *Adding Nodes Discovered from Traps on page 117*.

The NerveCenter node list contains two nodes with the same address but different names

Problem: The node name was changed. When a resynchronization occurred between NerveCenter and the network management platform, the platform added the node.

Solution: Change the name of the node in your platform, not in NerveCenter's node list.

See *Synchronization with Your Network Management Platform on page 115*.

NerveCenter deletes a node I added manually

Problem: At a resynchronization between NerveCenter and a network management platform, NerveCenter deletes any nodes marked autodelete that are not found in the platform's node database.

Solution: When you add the node disable the autodelete feature.

See *Adding and Deleting Nodes Manually on page 119*.

NerveCenter adds a node I deleted manually

Problem: At a resynchronization between NerveCenter and a network management platform or when NerveCenter runs the Discovery behavior model, NerveCenter adds any new nodes that fall within its filters.

Solution: Exclude the node in the IP filters before deleting it.

See *Filtering Nodes by IP Address on page 99*.

I’m seeing several errors recorded in the application event log window stating that ipsweep.exe is not running

Problem: Your system is looking for ipsweep.exe in the wrong directory.

Solution: Update the correct path to ipsweep.exe in the alarm action Command in the Discovery behavior model.

Problem: Ipsweep.exe is already currently running.

Solution: Kill the first process and restart the Discovery behavior model.

See *Using IPSweep Behavior Model in Designing and Managing Behavior Models* for more details.

Managing SNMP Settings

9

This chapter contains information you need to configure NerveCenter for SNMP communication and to support SNMPv3 agents.

This chapter includes the following sections:

Section	Description
Overview of NerveCenter SNMPv3 Support on page 126	Summarizes NerveCenter SNMPv3 support.
The Need for Node Classification on page 130	Discusses node classification and the ways in which NerveCenter classifies nodes.
SNMP Auto and Manual Classification Settings on page 133	Describes the different types of automatic and manual classification settings you can use.
Performance Tuning for Node Classification on page 138	Explains how to manage SNMP performance by controlling polling frequency and intervals.
Specifying NerveCenter SNMP Ports on page 143	Describes how to configure the ports NerveCenter uses to receive SNMP traps and send/receive SNMP correspondences.
Discovery and Initialization of SNMPv3 Agents on page 144	Summarizes the requirements for discovery and initialization of SNMPv3 agents.
Configuring an SNMPv3 Agent for NerveCenter on page 145	Summarizes the requirements for configuring an SNMPv3 agent for polling.
Configuring an Initial User for Discovering an SNMPv3 Agent on page 146	Summarizes how you configure an initial user without security restrictions so an SNMPv3 agent can be discovered.
Configuring SNMPv3 Security Settings on page 147	Describes how to change the authentication and privacy key passwords or the NerveCenter user and context.
Viewing the SNMPv3 Operations Log on page 149	Describes the Operations Log that records SNMPv3 operations and errors that occur while attempting to perform those operations.
SNMP Error Status on page 153	Describes SNMPv3 error status messages and indicates which ones cause polling to stop for a node.
SNMP Agent on page 157	Describes how to configure the NerveCenter SNMP Agent.

9 Managing SNMP Settings

Overview of NerveCenter SNMPv3 Support

NerveCenter support for SNMPv3 includes new data types and enhanced security for communication. SNMP v1 and v2c rely on community names for authentication. SNMPv3 enhances authentication and expands its services to include privacy. SNMPv3 also expands on the earlier concept of MIB views to control access to management information by using a View-based Access Control Model (VACM) to determine a user's access level for viewing MIB data.

Following are highlights of NerveCenter support for SNMPv3:

- Before NerveCenter can discover SNMPv3 agents on nodes, the nodes must have an initial user configured for discovery.

  See *Configuring an Initial User for Discovering an SNMPv3 Agent on page 146*.

  Refer to *Confirming the SNMP Version for a Node on page 125* for details about testing communication with a node using the NerveCenter Test Version poll.

- NerveCenter communicates (sends polls) with SNMPv3 agents on behalf of a specified NerveCenter user (the global User #1 or User #2 accounts, or a node-specific user). Before NerveCenter can poll SNMPv3 agents, the agents must be configured to support the user, security level, and potentially the context.

  See *Configuring an SNMPv3 Agent for NerveCenter on page 145*.

  See *Configuring SNMPv3 Security Settings on page 147*.

- NerveCenter supports three security levels for communicating with SNMPv3 agents. By default, NerveCenter sets the security level to noAuthNoPriv, which means the v3 agent sends and receives messages without authentication or encryption.

  See *NerveCenter Support for SNMPv3 Security on page 128* for details about security.

  Refer to *Changing the Security Level of an SNMPv3 Node on page 118* for details about setting a node's security level.

- The authentication and privacy protocols require specialized authentication and privacy keys, which are generated from the corresponding passwords. You can change these passwords in NerveCenter, thereby changing the keys.

  See *NerveCenter Support for SNMPv3 Digest Keys and Passwords on page 129*.

  See *Configuring SNMPv3 Security Settings on page 147*.

- NerveCenter supports the HMAC-MD5-96 (MD5) or HMAC-SHA-96 (SHA) protocols for authentication on a per-node basis, and DES, 3-DES, AES-128, AES-192, or AES-256 as privacy protocols. If you change an agent's authentication protocol, you must likewise configure NerveCenter to use that protocol to manage the corresponding node in its database.

Refer to *Changing the Authentication Protocol for an SNMPv3 Node on page 120* for details about changing the authentication protocol used by NerveCenter for an agent.

- A node must have SNMP version information before NerveCenter can poll or process a trap from that node. NerveCenter can discover the node version both automatically or manually. If auto-classification is enabled, then a newly added node (e.g., discovered from a trap, added from a platform such as HP OpenView, imported from another NerveCenter installation) will be classified at the highest level possible.

  NOTE
  NerveCenter auto-classification is disabled by default. You must enable it before NerveCenter can classify nodes added to its database.

  See *SNMP Auto and Manual Classification Settings on page 133*.

  Refer to *Classifying the SNMP Version Configured on Nodes in Designing and Managing Behavior Models* for details about classifying nodes manually.

- The trap source specified during installation can be changed to MSTrap, OVTrapD or NerveCenter. Changing the trap source requires stopping and starting the related applications (e.g., OVTrapD) and restarting the NerveCenter Server.

  See *Managing the NerveCenter Trap Source on page 52*.

- SNMPv3 operations are logged to a file so that you can follow the progress of v3 activities. The log includes information about activities (e.g., a key change initiated by the user) as well as errors that occur while NerveCenter attempts to perform the activities.

  See *Viewing the SNMPv3 Operations Log on page 149*.

  See *SNMP Error Status on page 153* for information about SNMPv3 errors.

- NerveCenter ships with behavior models that provide the status of various applications monitored by the SNMP Research CIAgent.

  For complete details about these and all behavior models, refer to the *Behavior Models Cookbook*.

NerveCenter Support for SNMPv3 Security

SNMPv3 enables two devices to communicate in a secure fashion using message authentication to validate users and encryption to ensure communication secrecy. SNMPv3 provides a User-based Security Model (USM) to establish authentication and secrecy.

SNMPv3 nodes can have one of the following security levels:

- **NoAuthNoPriv** — Neither message authentication nor encryption is used while communicating with the agent. No passwords are required.
- **AuthNoPriv** — Message authentication is used without encryption while communicating with the agent. An authentication protocol and password are required. The authentication password can be set to one of the global passwords defined under User #1 or User #2, or can be set on a per-node basis.
- **AuthPriv** — Both authentication and encryption are used when communicating with the agent. Both the authentication and privacy protocols and passwords are required. These passwords can be set to one of the global passwords defined under User #1 or User #2, or can be set on a per-node basis.

Communication between any two SNMPv3 entities takes place on behalf of a uniquely identified domain user. The security level used for this communication defines the security services — message authentication and encryption — used while exchanging data. NerveCenter communicates with SNMPv3 nodes on behalf of the NerveCenter poll user in the poll context.

If you do not specify a security level for an SNMPv3 node, NerveCenter uses the NoAuthNoPriv security level by default, which means that message authentication and encryption services are not used for data exchange with the node.

NOTE

The NerveCenter poll users (User #1 and User #2), contexts, and the authentication and privacy passwords can be changed in NerveCenter Administrator. If you change the passwords, you can update all nodes from the NerveCenter Administrator.

The node-specific information (such as version, security level, and authentication protocol) used to poll each SNMPv3 node is configured in NerveCenter Client. A node-specific poll user and associated credentials can be managed from the SNMP tab on the Node screen (see *Changing the Security Level of an SNMPv3 Node*).

NerveCenter Support for SNMPv3 Digest Keys and Passwords

SNMPv3 allows two devices to communicate in a secure fashion using message authentication and encryption to ensure secrecy. In any SNMPv3 communication, one of the two communicating entities plays a role of authoritative entity for the communication, and communication is performed on behalf of a unique user within the management domain.

The sender of a secure message attaches a code, or digest, for authentication and encrypts the message to ensure privacy. To generate this digest, the sender uses an authentication key at the authoritative entity of the user on whose behalf communication takes place. Similarly, to encrypt a message, the sender uses a privacy key at the authoritative entity of the user on whose behalf communication takes place. These keys are generated from the authentication password and privacy password, respectively.

SNMPv3 specifications have defined a localized key-generation scheme. For every user, the authentication key at every SNMPv3 entity is a function of the snmpEngineID of that entity, the user's authentication password, and the authentication protocol. For every user, the privacy key at every SNMPv3 entity is a function of the snmpEngineID of that entity, the user's privacy password, and the privacy protocol. NerveCenter supports this localized key-generation scheme.

NerveCenter communicates with SNMPv3 nodes on behalf of a NerveCenter user (User #1, User #2, or a local, node-specific user). NerveCenter needs to know the authentication and privacy passwords for this user to generate the keys required for secure communication. Whenever NerveCenter learns the snmpEngineID of a newly discovered SNMPv3 agent with a security level other than NoAuthNoPriv, NerveCenter generates these keys for the NerveCenter poll user on that agent:

- If authentication is required (a security level of AuthNoPriv is specified for the node), the sender uses the authentication key to generate the digest for the message, which is appended to the message.
- If encryption is required (a security level of AuthPriv is specified for the node), the sender uses the privacy key to generate the digest for the message. For this security level, only the privacy digest is required; privacy assumes authentication, and you cannot have encryption without authentication.

On receipt of a secure message, a receiver does the following:

- Separates the message from the digest (authentication or privacy).
- Uses the corresponding key from its local store to generate the message's local digest copy.
- Compares the local digest with the one received in the message. If the two digests match, the recipient authenticates or decrypts the message using the corresponding local key. If they do not match (indicating a lack of authentication), the recipient discards the message.
- The recipient reads and processes the message.

The Need for Node Classification

Each SNMP agent residing on a network device is represented by a corresponding node object in NerveCenter database. Agent attributes such as IP address, port number, and SNMP version are stored as properties of the corresponding node object in NerveCenter database. Whenever NerveCenter needs to communicate with any SNMP agent it monitors, it uses the required attributes from the corresponding node object in its database. When communicating with any SNMP agent, NerveCenter constructs an SNMP version-specific message based on the node's version information available in the NerveCenter database. Similarly, when receiving a message from a known agent, NerveCenter uses the version information available in the corresponding node object in its database to decide whether to process the message. To communicate with the agent on a managed device, NerveCenter must use the same SNMP version configured at the agent.

Many SNMP agent implementations support multiple versions of SNMP, as more complex enterprise networks are expected to have agents from multiple vendors supporting varied combinations of SNMP versions. A sophisticated network management product like NerveCenter, therefore, must provide a way to specify the SNMP version to be used when communicating with each of these agents. In very large networks, specifying this version information manually for each device quickly becomes an unmanageable task.

NerveCenter can classify nodes automatically based on SNMP version implementation on the corresponding agent, or you can manually classify nodes on a case-by-case basis. Finally, you can specify the SNMP version to use when NerveCenter communicates with the SNMP agent.

The NerveCenterNode Classification Algorithm

The NerveCenter node classification algorithm is depicted in *Figure 9-1*:

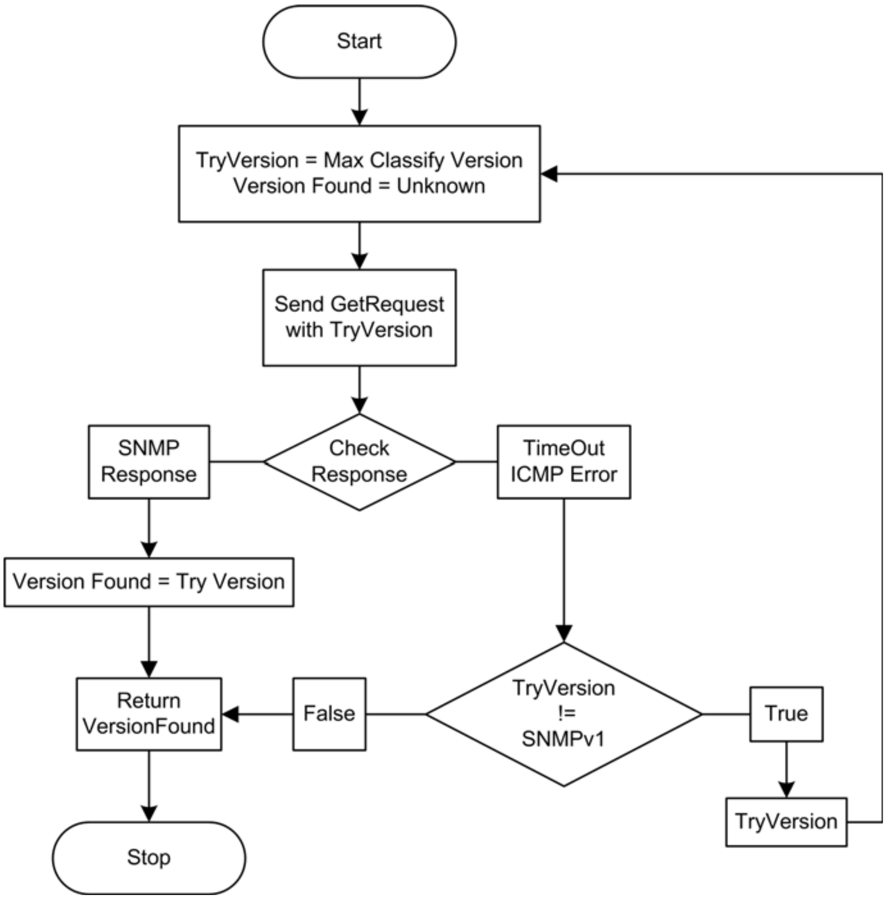


FIGURE 9-1. NerveCenter Node Classification Algorithm

How NerveCenter Classifies a Node

NerveCenter tries to classify a node based on the highest version of SNMP implemented on the corresponding SNMP agent. You can, however, limit the highest version NerveCenter attempts to classify in the NerveCenter Administrator. When you specify a maximum classification version, NerveCenter never attempts to classify a node above the version you specified.

When classifying a node, NerveCenter sends an SNMP GetRequest message and listens for the response. Receipt of an SNMP response or SNMP error indicates that the particular version is supported on the corresponding agent. If NerveCenter receives a response other than an SNMP reply, it concludes that the agent does not support that version. To ascertain support for SNMPv1 and SNMPv2c, NerveCenter sends a GetRequest message for SysObjectID with a read community of public.

To ascertain support for SNMPv3, NerveCenter sends a GetRequest message using the SNMPv3 configuration defined for the node. Thus if a node is defined to use SNMPv3 under AuthPriv with MD5 authentication, a digest key of “Kansas”, 3DES privacy, a passkey of “Dorothy”, a user name of “Toto” and an empty context field, all of that information is used in the GetRequest.

As of NerveCenter 5.1.04, there is no longer a default for the Classify operations’ SNMPv3 GetRequest. In prior releases a NoAuthNoPriv request was issued under the user name of “initial”.

Should the node not have a complete SNMPv3 definition, no SNMPv3 GetRequest is sent and the Classify operation moves to trying SNMPv2. A SNMPv3 configuration is complete if all of the following hold true:

- If the security level is NoAuthNoPriv, then a user name must be specified but the context may be empty.
- If the security level is AuthNoPriv, then the NoAuthNoPriv rules apply, an Authentication protocol must be selected, and the password must be filled in.
- If the security level is AuthPriv, then the AuthNoPriv and NoAuthNoPriv rules apply, the Privacy protocol must be selected, and the password must be filled in.

If NerveCenter does not get SNMP responses or SNMP errors to its GetRequest messages, then NerveCenter designates such nodes as ‘unknown’. For details of the algorithm NerveCenter uses to classify nodes, see *The NerveCenterNode Classification Algorithm on page 131*.

You can also test a particular version specified on a node by right-clicking a node in the NerveCenter Client node list view and selecting Test Version. This Test Version is similar to the GetRequest discussed above. If the agent does not support the version specified on the node, a test version failed error message is displayed.

SNMP Auto and Manual Classification Settings

NerveCenter can send requests (manually or automatically) to determine the highest SNMP version supported on a node. When you enable auto-classification, NerveCenter attempts to classify each node automatically when the node is added to the NerveCenter database.

A node must have correct version information, supplied manually or via auto-classification, before NerveCenter can poll or process traps from that node. See the following sections for more information:

- *How NerveCenter Classifies Node SNMP Versions on page 133*
- *When NerveCenter Classifies Node SNMP Versions on page 134*
- *Classifying Nodes Automatically on page 135*
- *Setting a Maximum Classify Value on page 137*

How NerveCenter Classifies Node SNMP Versions

There are two main ways that NerveCenter classifies nodes:

- **Manually** — You can issue a classify command in NerveCenter Client to classify one, several, or all nodes in the database. Procedures for issuing such commands are described in *Manually Changing the SNMP Version Used to Manage a Node on page 116*.
- **Automatically** — NerveCenter can be configured to classify nodes when they are added to its database (discovered from a trap, added from a platform such as OpenView Network Node Manager, or imported from another NerveCenter).

Each time NerveCenter attempts to classify a node, it sends a series of classification requests (GetRequest messages) to the node and classifies it based on the responses to those requests. Each request corresponds to an SNMP version — either v1, v2c, or v3.

While classifying a node, NerveCenter attempts to detect the maximum supported SNMP version on the agent up the limit configured in NerveCenter Administrator. For example, if you set a maximum classification version of v2c, NerveCenter never attempts to classify nodes any higher than v2c. (However, in the Client module, you can manually specify any version for a node and then test communication with the agent using that version.)

Based on the response to its messages, NerveCenter changes its SNMP version setting for the node.

CAUTION

Note the following about node classification:

- When NerveCenter attempts to classify a node, its node version information is lost.
- If NerveCenter fails to classify the node, then the node version is set to "Unknown." NerveCenter cannot poll a node with an unknown version.
- A node must have correct version information, either supplied manually or via auto-classification, before NerveCenter can poll or process a trap from the node.

When NerveCenter Classifies Node SNMP Versions

There are two main ways that NerveCenter classifies nodes:

- **On demand** — You can issue a classify command in NerveCenter Client to classify one, several, or all nodes in the database. Procedures for issuing such commands are described in *Configuring SNMP Settings for Nodes on page 115*.
- **Automatically** — You can set up auto-classification in NerveCenter Administrator. When NerveCenter adds nodes to its database (discovered from a trap, added from a platform such as OpenView Network Node Manager, or imported from another NerveCenter), any nodes without version information are classified at the highest possible level. NerveCenter does not attempt auto-classification for nodes that you add manually in Client.

When you enable auto-classification, NerveCenter attempts it in the following instances:

- A node is added through a node file from importutil or from the Client, and the node does not have a version or has the version "Unknown." This would happen, for example, if you were importing the node from a previous version of NerveCenter.
- A node is imported from a NerveCenter Server without a version or with version Unknown.
- A node is added from a trap, and the node's version is not v3. NerveCenter verifies whether these nodes are v1 or v2. If the trap is v3, NerveCenter does not need further verification.
- NerveCenter is co-resident with a network management platform that sends nodes to NerveCenter. All nodes added from OpenView Network Node Manager are v1 by default.

NOTE

NerveCenter does not attempt to auto-classify nodes that you add manually.

If you choose to disable auto-classification in the NerveCenter Administator, NerveCenter does not poll nodes whose SNMP version is unknown. (You can still classify nodes manually in NerveCenter Client with the available commands.)

Classifying Nodes Automatically

While you can always add and classify nodes manually, NerveCenter can automatically detect agents on network devices it monitors and add the corresponding node objects for those agents. Moreover, NerveCenter can be configured to recognize agents residing on network devices from other network management platforms like OpenView, another NerveCenter installation, and so on. In such circumstances, if the version information is not available for these automatically detected agents, NerveCenter assigns the version information to corresponding node objects as ‘unknown’. Since NerveCenter does not poll nodes having ‘unknown’ version, its default behavior is to automatically classify these automatically detected agents.

When NerveCenter’s autoclassification feature is enabled, NerveCenter attempts to autoclassify a node in the following situations:

- When nodes are imported via a node file (using ImportUtil or NerveCenter Client), and a node has no SNMP version information or the SNMP version is ‘unknown’. (Such a scenario might occur if you were importing nodes from a previous version of NerveCenter.)
- When nodes are imported from another NerveCenter Server, and a node has no SNMP version or the SNMP version of the node is ‘unknown’.
- When a new node is added to the NerveCenter database because a trap was received from a corresponding agent or source previously unknown to NerveCenter.
- When NerveCenter is co-resident with OpenView Network Node Manager and OpenView sends nodes to NerveCenter. In this case, NerveCenter does not know the SNMP version of the newly added node from OpenView.

NerveCenter auto-classification is enabled for whichever SNMP version you select, but NerveCenter never attempts to auto-classify a node for any version above your selection (see *Setting a Maximum Classify Value on page 137*). For example, if you select v2, NerveCenter can send classification requests for only v1 and v2.

Also, NerveCenter will not attempt to auto-classify a node using SNMPv3 if the node’s v3 definition is incomplete. The server recognizes the cases where a SNMPv3 request simply cannot succeed and in these cases the server skips the v3 testing of the node. Examples of this include a node with no SNMPv3 User Name – a blank user name is not allowed – or where Authentication is indicated but no digest value (passkey) is defined; for these cases the v3 check is skipped by the Classify operation.

NOTE

When NerveCenter receives SNMPv3 traps from unknown agents with an AuthNoPriv or an AuthPriv security level, those traps are processed on behalf of the poll user and in the configured poll context configured. (See the NerveCenter online help for more information.)

Follow the steps below to enable/disable the node auto-classification feature in the NerveCenter Administrator.

To enable or disable node auto-classification

NOTE
When auto-classification is disabled, then NerveCenter sets the nodes discovered by OpenView as SNMPv1.

1. In the NerveCenter Administrator, select the **Classify** tab.
2. To enable auto-classification, check the **Auto Classify** checkbox.

   NerveCenter auto-classification is enabled for whichever version you select, but NerveCenter never attempts to auto-classify a node for any version above your selection. For example, if you select v2, NerveCenter can send classification requests only for v1 and v2.
3. To disable auto-classification, uncheck the **Auto Classify** checkbox.

   NOTE
   If you disable auto-classification, bear in mind that NerveCenter does not poll nodes whose SNMP version is unknown.
4. Click **Save**.

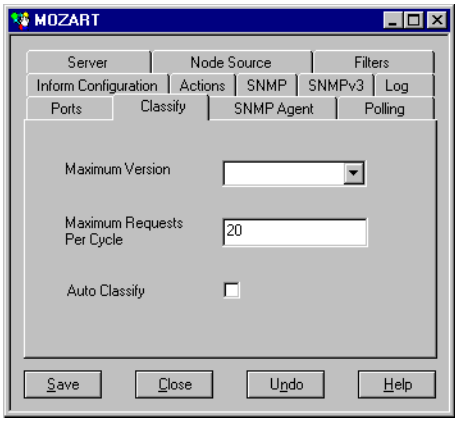
Setting a Maximum Classify Value

Whether using auto-classification or manual node classification (in Client), you specify the highest version you want NerveCenter to detect. When performing classification, NerveCenter will never attempt to classify a node for any version above the version you specify. For example, if you select v2c, NerveCenter can send classification requests only for SNMPv1 and v2c. (However, in the Client module, you can manually specify any version for a node and then test communication with the agent using that version.)

TO SET A MAXIMUM CLASSIFY VALUE

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server. See *Connecting Administrator to a NerveCenter Server on page 43*.
2. Select the **Classify** tab.

   The Classify tab is displayed.



Maximum Version

Maximum Requests Per Cycle 20

Auto Classify ☐

3. Select the highest level you want to classify from the **Maximum Version** list.
4. Click **Save**.

Performance Tuning for Node Classification

SNMPv3-specific operations for large numbers of nodes can cause bursts of traffic on the network. To help you to control this type of sudden SNMPv3-related rise of traffic on the network, you can set the number of requests per cycle and the poll interval.

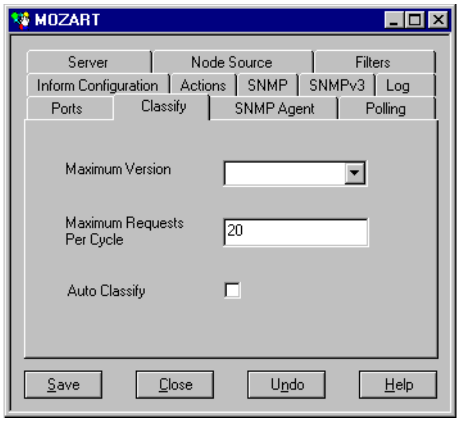
Setting the Maximum SNMPv3 Requests per Cycle

You control SNMPv3 performance by setting a maximum number of requests per processing cycle (approximately one second) for all v3 operations. Consider the number of SNMP messages per second your network devices can handle, and whether messaging uses authentication and privacy keys, which further slows performance.

To set the maximum number of SNMPv3 requests

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server.
2. Select the **Classify** tab.

   The Classify tab is displayed.



Maximum Version

Maximum Requests Per Cycle 20

Auto Classify ☐

3. In the **Maximum Requests Per Cycle** field, enter the maximum number of SNMPv3 requests you want NerveCenter to process. The default value is 20.
4. Click **Save**.

Specifying SNMP Poll Intervals for NerveCenter

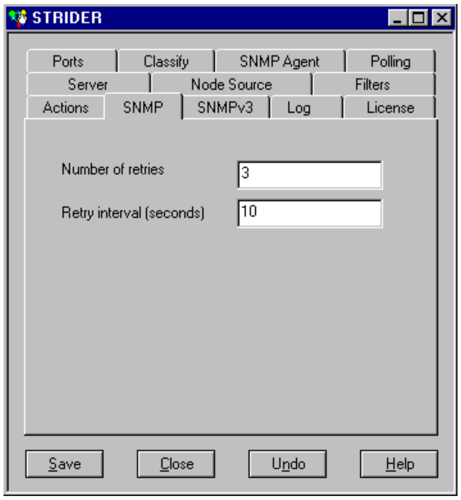
You can change the interval (three attempts by default) at which NerveCenter sends polls to SNMP agents to obtain MIB information. For example, NerveCenter would make four polling attempts to a device that could not respond, the initial attempt and three retries.

The retry interval is significant in that the rate should be high enough to account for the number of retries per interval. For example, if a NerveCenter Server is set to retry three times at a 30 second interval, the polling rate for any of its behavior model's polls should be no lower than 90 seconds.

To specify NerveCenter poll settings

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server.
2. Select the **SNMP** tab.

   The SNMP tab is displayed.



Number of retries: 3

Retry interval (seconds): 10

3. In the **Number of retries** field, enter the number of times to reissue unanswered SNMP or ICMP request polls.
4. In the **Retry interval** field, enter the number of seconds NerveCenter should wait for a reply to a poll before issuing another.
5. Select **Save**.

NerveCenter will handle SNMP data according to these settings.

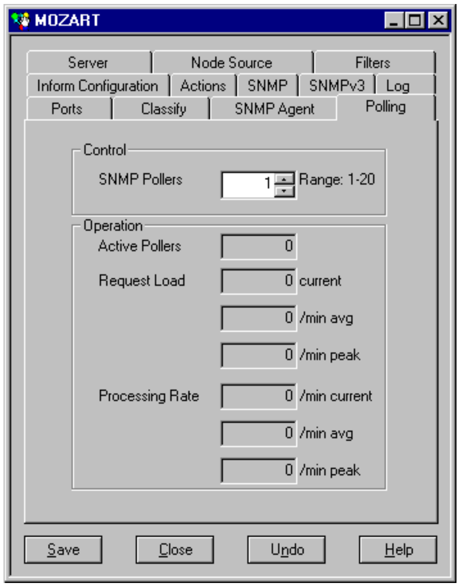
Multi-Threaded Polling

NerveCenter 5.1 introduces support for multi-threaded polling, which can significantly improve performance. Multi-threaded polling is a licensed feature; the License tab of the Server Status window (see page 141) displays the number of threads you can use.

TO CONFIGURE MULTI-THREADED POLLING

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server.
2. Select the **Polling** tab.

   The Polling tab is displayed.



MOZART

Control
SNMP Pollers: 1 — Range: 1-20

Operation
Active Pollers: 0
Request Load: 0 current; 0 /min avg; 0 /min peak
Processing Rate: 0 /min current; 0 /min avg; 0 /min peak

3. Specify the number of polling threads to run in parallel in the **SNMP Pollers** field.
4. Click **Save**.

The Operation section of the Polling tab displays information calculated by the ncserver process and sent to ncadmin. The values are for the entire running of ncserver and are independent of when the ncadmin process connects to the server. (i.e., they are for the life of the ncserver). *Table 9-1* describes the displayed data.

TABLE 9-1. Polling Performance Data

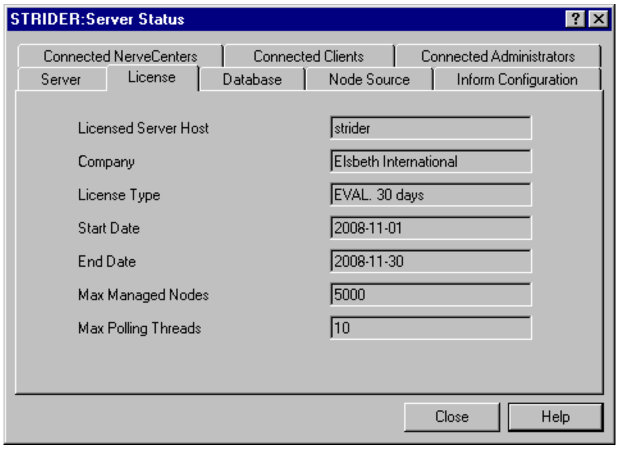
Field		Description
Active Pollers		Displays the number of actively running pollers. This field is useful while system is starting up or adjusting polling rate.
Request Load	current	The number of requests currently being processed by active pollers.
	/min avg	The average number of requests being processed.
	/min peak	The maximum number of requests being processed.
Processing Rate	/min current	The number of requests processed over the last minute.
	/min avg	The average per-minute processing rate.
	/min peak	The maximum per-minute processing rate.

You can view the number of available threads from the License tab of the Server Status window.

TO DISPLAY INFORMATION ABOUT THE ACTIVE SERVER'S NERVECENTER LICENSE

1. Choose **Server Status** from the **Server** menu.
2. Click the **License** tab.

   The License tab is displayed.



Licensed Server Host	strider
Company	Elsbeth International
License Type	EVAL. 30 days
Start Date	2008-11-01
End Date	2008-11-30
Max Managed Nodes	5000
Max Polling Threads	10

The License tab displays a server’s NerveCenter license details; these settings are configured in the hostname.dat license file on the server and cannot be changed here. *Table 9-2* describes the fields on the License tab:

TABLE 9-2. Server Status - License Tab Fields

Field	Description
Licensed Server Host	The Server for which the license information is being displayed.
Company	The license owner.
License Type	Indicates if this server is using a standard or evaluation license.
Start Date	Starting date when the license can be used.
End Date	Termination date after which the license is no longer valid.
Max Managed Nodes	The maximum number of nodes that can be managed from this NerveCenter Server.
Max Polling Threads	The number of poll threads that can run in parallel, if the Multi-Threaded Polling feature has been licensed.

Specifying NerveCenter SNMP Ports

NerveCenter has two primary sources of information about network conditions:

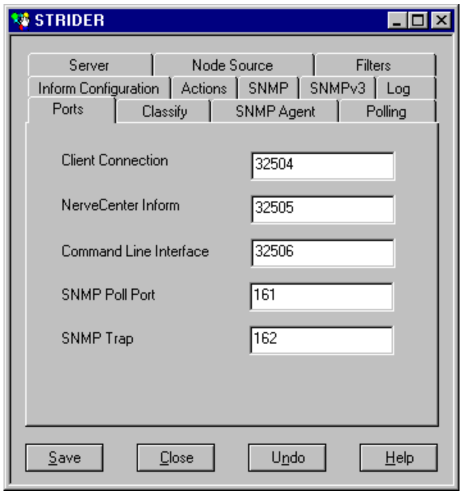
- NerveCenter listens passively for SNMP traps sent by a managed device.
- NerveCenter actively polls the SNMP agents on a managed device.

You can change the ports that NerveCenter uses for receiving traps and sending polls.

To specify SNMP port settings for NerveCenter

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server. See *Connecting Administrator to a NerveCenter Server on page 43*.
2. Select the **Ports** tab.

   The Ports tab is displayed.



Client Connection 32504

NerveCenter Inform 32505

Command Line Interface 32506

SNMP Poll Port 161

SNMP Trap 162

3. In the **SNMP Poll** field, enter the number of the port you want NerveCenter to use to communicate with SNMP agents. This port is used to get or set SNMP information.

   The value entered here specifies the port on the node to which NerveCenter sends SNMP polls. You can change the port for any particular node in the node's definition window in NerveCenter Client. It is suggested that you keep the default port number 161.
4. In the **SNMP Trap** field, enter the port number that you want NerveCenter to use for receiving SNMP traps. This setting has no effect if NerveCenter is co-resident with HP OpenView Network Node Manager. OpenView has the trap port and an internal process enables NerveCenter to receive a copy of the trap. The default port number is 162.

5. Click **Save**.

NOTE
You must shut down and restart the NerveCenter Server before the SNMP Trap port change takes effect. To capture SNMPv3 traps, the SNMP Trap port must be free of traps, which requires you to shut down MS Trap service and OVTrapD.

Discovery and Initialization of SNMPv3 Agents

NerveCenter can discover and monitor nodes with SNMPv1 and v2 agents without knowing anything in particular about the agent beforehand. For nodes with SNMPv3 agents, however, NerveCenter requires certain information before the node can be discovered and managed. The process of obtaining this information is call initialization.

Initialization occurs in two cases:

- After installation to obtain engine information required for polling.
- When a node’s version is changed to SNMPv3 from another version.

Before NerveCenter can initialize a node, its SNMPv3 agent must be configured with the poll user, poll context, and security level.

- See *Configuring SNMPv3 Security Settings on page 147* for information about the poll user, poll context, and security level.
- See *Configuring an SNMPv3 Agent for NerveCenter on page 145* for configuration details.

Initialization consists of the following:

1. Obtaining the agent’s snmpEngineID value for SNMPv3 node discovery. See *Configuring an Initial User for Discovering an SNMPv3 Agent on page 146* for details.
2. Obtaining additional engine information if required for the chosen level of security.

   NerveCenter requires additional engine information when the NerveCenter user has a security level other than NoAuthNoPriv. In this case, NerveCenter must obtain SNMPv3 agent’s snmpEngineBoots and snmpEngineTime values.

   NOTE
   SNMPv3 specifications allow secure initial agent configuration wherein an agent need not disclose its snmpEngineID. Network management applications must obtain on their own the snmpEngineID for such agents. NerveCenter cannot initialize or poll such agents.

Configuring an SNMPv3 Agent for NerveCenter

Before NerveCenter can poll SNMPv3 agents, those agents must be configured to support one of the two global NerveCenter users (User #1 or User #2), or configured for a node-specific local user.

User #1 and User #2 definitions are set up through the NerveCenter Administrator. The values supplied for these two definitions are called "global" because nodes that are configured to use either of these two definitions *share* these values — a change to any field within User #1 or User #2 is immediate and effective on *all* nodes that are configured to use that definition.

As an alternative to this shared, global usage mechanism, nodes can be configured with a *local* SNMPv3 definition. The NerveCenter Client is used to set which SNMPv3 definition it will use (User #1, User #2, or Local) and to set the values for a Local definition. Nodes in Local mode have their own private SNMPv3 attributes.

Nodes that reference User #1 or User #2 can be set to access a given security level from the chosen global definition. SNMPv3 polls are formed by NerveCenter using as much of the definition that applies to the selected security level set for each node.

Two passwords (authentication and privacy) and a context can be configured for each NerveCenter user (#1, #2, and Local). You must supply the authentication and privacy passwords for the NerveCenter user configured on your devices. .

Table 9-3 describes the configuration options for SNMPv3 security.

TABLE 9-3. SNMPv3 Agent Configuration

Field	Description
User #1 User User #2 User	String containing the name of the SNMPv3 user account referred to as the global User #1 or User #2 accounts.
The following items can be configured for each user:	
Authentication Protocol	Authentication Protocol (MD5 or SHA-1) for each user.
Authentication Key	Click **Set Auth Key** to enter the password to generate an Authentication Key.
Privacy Protocol	Privacy Protocol (DES, 3-DES, AES-128, AES-192, or AES-256) for each user.
Privacy Key	Click **Set Priv Key** to enter the password to generate a Privacy Key.
Context	This field contains the context string for the account. This information is required only if a context is defined on the SNMPv3 device.

NerveCenter then polls the SNMPv3 agent on behalf of the poll user in this context. In addition to the poll user and context, the parameters described in the following sections must be configured on the SNMPv3 agent.

NOTE
The following descriptions serve only as guidelines for configuring SNMPv3 agents for NerveCenter. For exact procedural details about configuring SNMPv3 agents, consult the user documentation supplied by your SNMPv3 agent provider.

Configuring an Initial User for Discovering an SNMPv3 Agent

To discover an SNMPv3 node, NerveCenter sends a GetRequest message to that node with the following parameters:

- SecurityLevel: set to (noAuthNoPriv)
- MsgUserName: set to the user name provided by the node's SNMPv3 configuration
- MsgAuthoritativeEngineID: set to zerolength (' ')
- Empty variable bindings (variable-bindings=' ')

NOTE
For exact procedural details about configuring SNMPv3 agents, consult the user documentation supplied by your SNMPv3 agent provider.

Properly configured SNMPv3 agents respond to GetRequest with a Report PDU, which includes its local snmpEngineID (supplied within the msgAuthoritativeEngineID field). After NerveCenter obtains the snmpEngineID, NerveCenter can discover the node as an SNMPv3 agent.

This process of discovering SNMPv3 agents is recommended in SNMPv3 specifications. Some SNMPv3 agents are preconfigured with an "initial" user and respond to the GetRequest as described without any special type of configuration — see documentation supplied by the SNMPv3 agent provider for detailed information.

When a node is created in NerveCenter (via the NerveCenterAdministrator application), its SNMPv3 "Select User" toggle is set to "User #1" by default. Thus, for most incoming nodes it will be the case that the discovery and classify operations will attempt to use the User Name field from User #1 to contact the SNMP Agent to see whether it can get an snmpEngineID.

If User #1 has an empty User Name field, then the NerveCenter server simply skips any SNMPv3 discovery on the node because SNMPv3 does not allow an empty MsgUserName field to be sent. If the user puts the word "initial" into the User #1 User Name field, then that would be the value attempted for cases where a node is newly received and a discovery or classify is attempted.

NerveCenter can communicate with devices that support any available version of SNMP: v1, v2c, or v3. All SNMP agents and network devices managed by NerveCenter are represented as nodes in the database; the database also stores the SNMP version assigned to each node. Such SNMP version information can be assigned manually or automatically.

Configuring SNMPv3 Security Settings

Before NerveCenter can poll SNMPv3 agents, the agents must be configured to support a NerveCenter user. There are two global accounts, User #1 and User #2 that can be configured within NerveCenter. You can also define accounts on a per-node basis from the Node screen in NerveCenter Client.

After installation, you can change the passwords as appropriate for your network management strategy. Each NerveCenter user can be assigned authentication and privacy protocols, each with its own password-generated key. The security level determines whether passwords are needed:

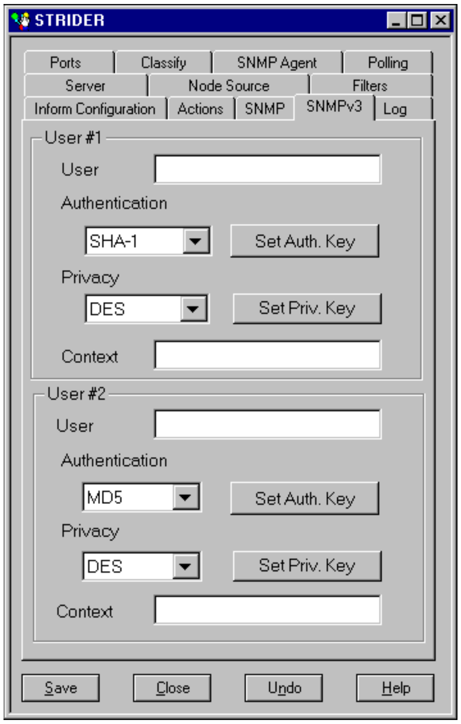
- Authentication password is required for AuthNoPriv and AuthPriv security.
- The Privacy password is required for AuthPriv security.

NerveCenter itself does not require an SNMPv3 context; you may need to enter it to support individual agents that may require it. If you change these values on your agents, you must make the same change in NerveCenter Administrator. If you use contexts, changing context on a regular basis can help ensure privacy; also, if you have multiple NerveCenters managing different device categories, you can associate different user names and contexts with each category.

To configure SNMPv3 security

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server. Select the **SNMPv3** tab.

   The SNMPv3 tab is displayed.



User #1
User
Authentication: SHA-1
Privacy: DES
Context

User #2
User
Authentication: MD5
Privacy: DES
Context

2. Edit the **User #1** or **User #2** account names if necessary.
3. Configure the following for each user as necessary:
   a. Select the **Authentication** protocol (MD5 or SHA-1).
   b. Click **Set Auth. Key**, enter the password in each field, and click **OK**.
   c. Select the **Privacy** protocol (DES, 3DES, AES-128, AES-192, or AES-256).
   d. Click **Set Priv. Key**, enter the password in each field, and click **OK**.
   e. If an agent configuration requires it, enter the **Context** for the account.
4. Click **Save**.

Viewing the SNMPv3 Operations Log

Whenever a NerveCenter Server receives a request for an SNMPv3 operation (e.g., authorization or privacy key change request) or an error occurs while attempting to perform an SNMPv3 operation (e.g., v3 initialization fails), the NerveCenter Server logs a message to V3Messages.log, which resides in the NerveCenter installation log directory on the NerveCenter Server host machine. The file contains messages about SNMPv3 operations and errors resulting from requests that originate with any connected NerveCenter Clients, Administrators, and Command Line interfaces.

After logging the error, the NerveCenter Server notifies all connected NerveCenter Clients and Administrators in the following ways:

- If you are logged on to the NerveCenter Client or Administrator that initiated the operation that caused an error condition, NerveCenter displays the error that was logged.
- If you are logged on to a NerveCenter Client or Administrator that did not initiate the error condition, a red icon appears in the status bar; double-click the icon to display the NerveCenter Server with the SNMPv3 error. If your Client or Administrator is connected to more than one Server, the dialog box lists all servers that currently have an error condition.

NOTE
The dialogs are displayed only in the NerveCenter Client, not the NerveCenter Web Client.

When your NerveCenter Client or Administrator displays a dialog box with an error condition, you can do either of the following:

- Acknowledge the error condition by “signing the log.” When you sign the log, NerveCenter notes that in the log file and changes the red icon back to green for all connected Clients and Administrators.
- Dismiss the dialog box without acknowledging the error condition, in which case only the icon in your Client or Administrator turns green. The icon remains red for all other connected Clients and Administrators to signal that the NerveCenter Server has an unacknowledged/unsigned error. Moreover, the Server does not indicate acknowledgment in the log file.

If the SNMPv3 operation affects a group of nodes (e.g., a version change or classification failure), only one error instance for the group displayed; see the log file for details on individual nodes.

You must have administrator rights to initiate an SNMPv3 operation that can result in an error or to acknowledge a logged error condition. If you are logged on with only user rights, you can dismiss the error dialog box but not acknowledge an error condition.

Whether you acknowledge or dismiss the error, all messages remain in the V3Messages.log.

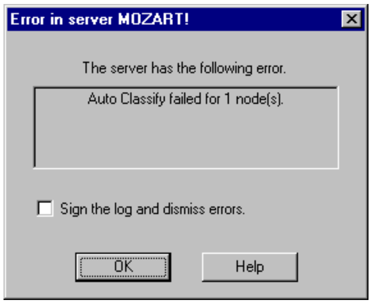
For more information, refer to the following topics:

- *Signing a Log for SNMPv3 Errors Associated with Your Administrator on page 150*
- *Signing a Log for SNMPv3 Errors Associated with a Remote Client or Administrator on page 151*
- *Viewing the SNMPv3 Operations Log on page 152*

Signing a Log for SNMPv3 Errors Associated with Your Administrator

Whenever an SNMPv3 operation is requested or an error occurs while attempting an SNMPv3 operation, the NerveCenter Server logs a message to V3Messages.log. If you are logged in to the NerveCenter Client that initiated the logged request, NerveCenter displays a dialog box with that error.

a


Error in server MOZART!

The server has the following error.

Auto Classify failed for 1 node(s).

☐ Sign the log and dismiss errors.

FIGURE 9-2. Operations Log Error in Server Dialog Box for Your Client

Users with administrator rights can acknowledge a logged condition from NerveCenter Client by signing the Operations log. Signing the log causes the icon to turn green in all connected Clients/Administrators.

You can also dismiss the dialog box without acknowledging the error condition. If you are logged on with user rights rather than administrator rights, your only option is to dismiss the dialog box; you cannot sign the Operations log.

To sign the Operations log

1. After viewing the message that NerveCenter displays on your screen, check the Sign the log and dismiss errors checkbox.
2. Click **OK**.

The icon in the Status Bar turns green for all Clients or Administrators connected to the designated NerveCenter Server. You can later view this message again in the Operations log.

This V3Messages.log file, resides in the NerveCenter installation log directory. The file can be viewed in a text editor or word processor.

To dismiss the Error in Server dialog box

- Click **OK** without checking the checkbox.

  In this case, only the icon in your Client turns green. For all other connected Clients and Administrators, the icon remains red and signals to those modules that the NerveCenter Server has some error that remains unacknowledged.

Signing a Log for SNMPv3 Errors Associated with a Remote Client or Administrator

Whenever an error occurs while attempting an SNMPv3 operation, the NerveCenter Server logs a message to V3Messages.log. If you are logged on to a remote NerveCenter Client (one that did not initiate the error condition), the status bar displays a red icon.

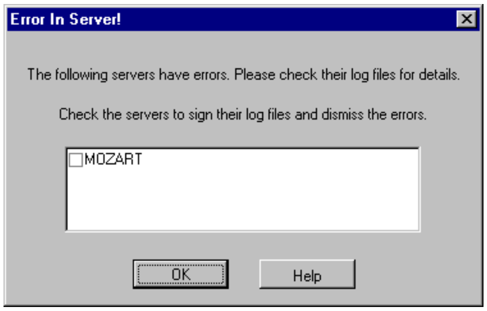
Users with administrator rights can acknowledge a logged condition from NerveCenter Client by signing the Operations log. Signing the log causes the status icon to turn green in all connected Clients/Administrators.

You can also dismiss the dialog box without acknowledging the error condition. If you are logged on with user rights rather than administrator rights, your only option is to dismiss the dialog box; you cannot sign the Operations log.

To sign the Operations log

1. Double-click the red icon in the Status Bar.

   The Error In Server dialog box is displayed.


Error In Server!

The following servers have errors. Please check their log files for details.

Check the servers to sign their log files and dismiss the errors.

☐ MOZART

2. Check the NerveCenter Server or Servers for which you want to sign the log.
3. Click **OK**.

   The icon in the Status Bar turns green for all Clients or Administrators connected to the servers you checked. At a suitable time, you can open the Operations log and view the new message. This file, named V3Messages.log, resides in the NerveCenter installation log directory. The file can be viewed in a text editor or word processor.

To dismiss the Error In Server dialog box

1. Double-click the red icon in the Status Bar.

   The Error In Server dialog box is displayed.
2. Click **OK** without checking any of the checkboxes.

   In this case, only the icon in your Client turns green. For all other connected Clients and Administrators, the icon remains red and signals to those modules that the NerveCenter Server has some error that remains unacknowledged.

Viewing the SNMPv3 Operations Log

Whenever an SNMPv3 operation is requested or an error occurs while attempting the operation, the NerveCenter Server logs a message to the V3Messages.log file, which resides in the NerveCenter installation log directory on the NerveCenter Server host machine.

The file can be viewed in a text editor or word processor. As NerveCenter adds more messages to the file, the file continues to grow until you manually remove old messages.

The log entries resemble the following:

```
06/20/2000 09:26:29 Tue - Event ID : NC_SERVER; Category ID :
NC_THREAD_V3OP;Error Status : AutoClassifyFail; Error while communicationg
using SNMPv1 for 10.52.174.51 because of : NC_PORT_UNREACHABLE;
```

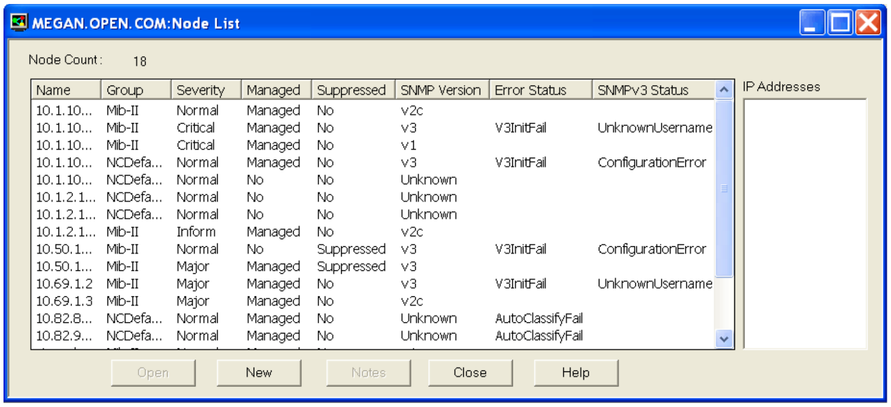
Following are the fields in the log:

TABLE 9-4. Fields in the Operations Log

Field	Description
Date/Time	Date and time the record was logged. The format is month/day/year, hour/minute/second, and day (for example, 12/16/2000 11:32:29 Sat).
EventID	This always NC_SERVER.
CategoryID	Name of the thread where the event occurred.
Error Status	One of several error status strings. See *Error Status* for a description of SNMPv3 error status messages and which ones cause polling to stop for a node.
Error Description	Details of the error or operation.

SNMP Error Status

When NerveCenter is unable to complete an SNMP operation on a node, the error status is displayed in the Node List (NerveCenter Client and Web Client) and in the SNMP tab of the node's definition window (NerveCenter Client). *Figure 9-3* shows the Node List window in the Client.


MEGAN.OPEN.COM:Node List

Node Count: 18

Name	Group	Severity	Managed	Suppressed	SNMP Version	Error Status	SNMPv3 Status
10.1.10...	Mib-II	Normal	Managed	No	v2c		
10.1.10...	Mib-II	Critical	Managed	No	v3	V3InitFail	UnknownUsername
10.1.10...	Mib-II	Critical	Managed	No	v1		
10.1.10...	NCDefa...	Normal	Managed	No	v3	V3InitFail	ConfigurationError
10.1.10...	NCDefa...	Normal	No	No	Unknown		
10.1.2.1...	NCDefa...	Normal	No	No	Unknown		
10.1.2.1...	NCDefa...	Normal	No	No	Unknown		
10.1.2.1...	Mib-II	Inform	Managed	No	v2c		
10.50.1...	Mib-II	Normal	No	Suppressed	v3	V3InitFail	ConfigurationError
10.50.1...	Mib-II	Major	Managed	Suppressed	v3		
10.69.1.2	Mib-II	Major	Managed	No	v3	V3InitFail	UnknownUsername
10.69.1.3	Mib-II	Major	Managed	No	v2c		
10.82.8...	NCDefa...	Normal	Managed	No	Unknown	AutoClassifyFail	
10.82.9...	NCDefa...	Normal	Managed	No	Unknown	AutoClassifyFail	

FIGURE 9-3. Node List Window

Though most of the error strings correspond to SNMPv3 errors, some are applicable for v1 and v2c errors as well. These are noted in the descriptions below.

Sometimes error conditions can be corrected simply by running the SNMP Test Version poll. Others may require configuration changes to the node's SNMP agent. After changing the configuration of an SNMP agent, always test communication with the node in NerveCenter Client prior to polling the node.

The following list describes each possible SNMP error status.

- **V3InitFail** – An attempt to get the snmpEngine ID of an SNMPv3 agent failed or the SNMPv3 configuration defined for that node is causing a failure at the SNMPv3 communication layer. This can occur either when NerveCenter first attempts to poll the node using the SNMPv3 configuration or at any point when the SNMP agent changes its SNMPv3 configuration. For all of these cases, the V3InitFail is augmented by one of the following values in the SNMPv3 Status field (NerveCenter Client):
    - ConfigurationError – The node’s SNMP definition is incomplete with respect to its Security Level. This status is discovered and reported by NerveCenter before issuing an SNMPv3 request to an SNMP Agent.

      Operator intervention is required. The node’s SNMP v3 definition must contain a User Name regardless of the Security Level — AuthNoPriv requires an Authentication Protocol and Password; AuthPriv requires Authentication and Privacy Protocols and passwords for each.
    - UnknownUsername – The SNMP Agent reports that the SNMPv3 User Name being sent by NerveCenter is not one of the user names that it has been configured to handle.
    - UnknownContext – The SNMP Agent reports that the SNMPv3 Context being sent by NerveCenter is not appropriate. Many SNMP Agents do not report this value, even if it is the underlying issue. Instead, the SNMP Agent may not issue any response and the operation will time out.
    - UnavailableContext – The SNMP Agent reports that the SNMPv3 Context being sent by NerveCenter is known but inapplicable to the operation (poll, discovery, or classification) being attempted. Many SNMP Agents do not report this value, even if it is the underlying issue. Instead, the SNMP Agent may not issue any response and the operation will time out.
    - UnsupportedSecLevel – The SNMP Agent reports that it cannot handle the Security Level defined in a request issued to it by NerveCenter.

  - UnknownEngineID – Either NerveCenter's SNMP Stack or the SNMP Agent is reporting an issue with the snmpEngineID used for SNMP v3 communication. This can occur if the snmpEngineID is changed on the SNMP Agent during polling.
  - IncorrectAuthPasskey – The SNMP Agent reports that the Authentication passkey (digest) being issued by NerveCenter is not correct. This generally occurs in one of two cases: 1) An incorrect password was entered either on the SNMP Agent or in NerveCenter, or 2) The password was entered correctly at both ends, but the selected Authentication protocol is mismatched between the SNMP Agent and NerveCenter.
- **ClassifyFail** – An attempt to obtain the node's SNMP version failed during a classification attempt. The node's version will be set to "Unknown" and it will not be polled. You can manually change the version or try to classify the node again.
- **AutoClassifyFail** – An auto-classification attempt failed to obtain the node's version. The node's version will be changed to "Unknown" and it will not be polled. You can manually change the version or try to classify the node again.

  NOTE

  ClassifyFail and AutoClassifyFail status values are not limited to SNMPv3 agents. If NerveCenter attempts to classify an agent and fails for some reason (e.g., the agent is down), NerveCenter will mark the node with ClassifyFail or AutoClassifyFail regardless of the SNMP version supported on the agent.

- **TestVersionFail** – At attempt to poll the SNMP agent failed. The Test Version poll sends a GetRequest message for a node based on the SNMP version configured for that node.

  If the Test Version poll fails, polling will not happen for this node. In that case, you may need to reconfigure the agent on this node. Then, try running the Test Version poll again (from a node's definition window or the right-click menu in the node list).

  NOTE

  TestVersionFail is not limited to SNMPv3 agents. You can test the version of any SNMP agent with this feature.

- **Configuration Mismatch** – Indicates an SNMP trap was received but there is some problem with the configuration on the agent. If NerveCenter is unable to decode a trap due to some unspecified reason (e.g., unsupported authentication or privacy parameters on the agent or an incorrect NerveCenter user name), NerveCenter can receive the trap and add the node to its database if configured to discover nodes via traps. After adding the node to its database, however, NerveCenter assigns an error status of Configuration Mismatch.

NOTE
Any error that occurs during trap decoding always results in a Configuration Mismatch error.

- **TimeSyncFail** – An attempt to get the node’s snmpEngine boots/timeticks failed. Polling will continue for this node. If any polls successfully reach the node, the node responds with an “Out of time window” report PDU that contains the correct boots/timeticks, and NerveCenter can then update this information for the node. For the initial polls that generate the report PDU, the SNMP_NOT_IN_TIME_WINDOW trigger will be fired.
    - You can ignore this message, which simply indicates that NerveCenter is getting in sync with that node. You can recover from this error status by right-clicking the node in the Node List and selecting v3TestPoll. If the agent corresponding to the node is up, the test poll should be successful and clear the error message. The SNMPv3 Status field will be set to the following:
        - NotInTimeWindow – This is the reply sent by the SNMP Agent or declared by NerveCenter’s SNMP stack upon investigating a request or response PDU wherein the SNMPv3 timestamp handling shows a time sync failure.

SNMP Agent

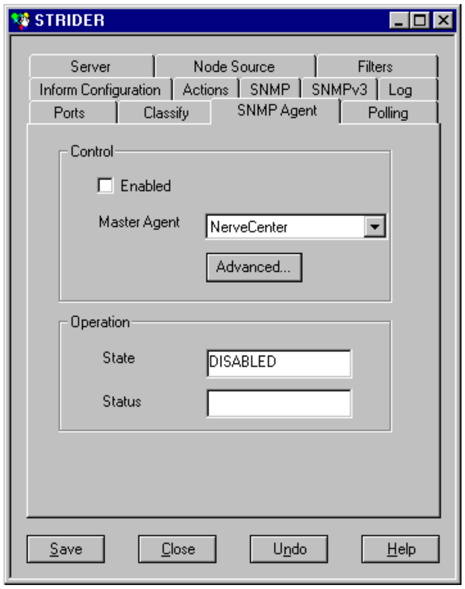
NerveCenter includes a Master SNMP Agent and an SNMP Sub-Agent so that you can monitor NerveCenter with a third party SNMP application, such as HP OpenView. The data provided by the NerveCenter SNMP Agent is described in the MIB modules located in *InstallPath*/mibs/nsmg.

When you install NerveCenter, you also install the SNMP Master and Sub Agents, however they are disabled by default. You need to first configure the SNMP Agent and then enable it before it can process SNMP requests. When you enable the SNMP Agent for the first time, the Agent is added to the set of Services or Daemons running on the host system.

To configure the NerveCenter SNMP Agent

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server. See *Connecting Administrator to a NerveCenter Server on page 43*.
2. Select the **SNMP Agent** tab.

   The SNMP Agent tab is displayed.



STRIDER

Tabs: Server, Node Source, Filters, Inform Configuration, Actions, SNMP, SNMPv3, Log, Ports, Classify, SNMP Agent, Polling

Control

☐ Enabled

Master Agent: NerveCenter

Advanced...

Operation

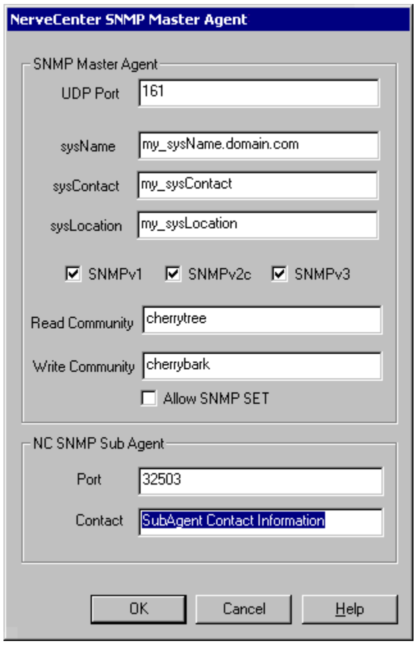
State: DISABLED

Status:

Save · Close · Undo · Help

3. Select NerveCenter from the Master Agent list.
4. Click **Advanced**.

The NerveCenter Master Agent dialog box is displayed.


NerveCenter SNMP Master Agent

SNMP Master Agent

UDP Port: 161

sysName: my_sysName.domain.com

sysContact: my_sysContact

sysLocation: my_sysLocation

[✓] SNMPv1 [✓] SNMPv2c [✓] SNMPv3

Read Community: cherrytree

Write Community: cherrybark

[ ] Allow SNMP SET

NC SNMP Sub Agent

Port: 32503

Contact: SubAgent Contact Information

NOTE

You can edit the NerveCenter Master Agent dialog box if the SNMP Agent is disabled. See *To disable the NerveCenter SNMP Agent on page 159*.

5. Enter the **UPD Port** from which you want the Master Agent to send SNMP messages.
6. Enter the **sysName** (system name), **sysContact** (system contact), and **sysLocation** (system location) in the appropriate fields.
7. Select the versions of SNMP you want to support.
8. If you select SNMPv1 or SNMPv2c, enter **Read Community** and **Write Community** strings.

NOTE

SNMP Set is disabled for this release of NerveCenter.

If you select SNMPv3, use the security settings defined on the SNMPv3 tab. See *Configuring SNMPv3 Security Settings on page 147* for more details.

9. Click **OK**.

   You return to the SNMP Agent tab.

10. Select the **Enabled** check box.

    NOTE
    To enable the SNMP Agent on UNIX, the NerveCenter server must be running as a process with root privileges.

11. Click **Save**.

If you want to limit network traffic, or have no more need of the SNMP Agent, you can turn it off.

NOTE
To disable the SNMP Agent on UNIX, the NerveCenter server must be running as a process with root privileges.

TO DISABLE THE NERVECENTER SNMP AGENT

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server. See *Connecting Administrator to a NerveCenter Server on page 43*.
2. Select the **SNMP Agent** tab.

   The SNMP Agent tab is displayed.
3. Clear the **Enabled** check box.
4. Click **Save**.

In addition to the SNMP Master Agent configuration, you must configure the settings between the SNMP Sub Agent and the NerveCenter server.

To configure communication between the SNMP Sub Agent and NerveCenter Server

Note

You can only make changes to the NerveCenter Master Agent dialog box if the SNMP Agent is disabled. See *To disable the NerveCenter SNMP Agent on page 159*.

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server. See *Connecting Administrator to a NerveCenter Server on page 43*.
2. Select the **SNMP Agent** tab.

   The SNMP Agent tab is displayed.
3. Click **Advanced**.

   The NerveCenter Master Agent dialog box displays.
4. Enter the TCP **Port** number for the communication between the SNMP Sub Agent and the NerveCenter Server.

   The default value for this field is 32503.
5. Enter the contact information for the SNMP Sub Agent.

   Use this field to indicate the person(s) responsible for the NerveCenter service on this host.
6. Click **OK**.

   You return to the SNMP Agent tab.
7. Click **Save**.

Managing ICMP Settings

10

This chapter explains how NerveCenter uses ICMP and the controls available for configuring and monitoring this protocol within NerveCenter.

Overview of ICMP Support

ICMP is supported in NerveCenter as a protocol that can be utilized for both monitoring and polling. ICMP, defined in RFC792, is an internal communication protocol embedded within the IP layer of the TCP/IP protocol suite. It is most often employed by internal elements for configuration modification.

The following are the features available for accessing ICMP through NerveCenter

- Nodes can be “pinged” (ICMP Echo / Echo Response) by NerveCenter. The response, or lack thereof, can be used to drive NerveCenter models in the same way done with SNMP polling.
- ICMP messages can be monitored and filtered in the same manner as SNMP Traps. Through the use of NerveCenter Trap Masks, triggers can be created to drive NerveCenter models in the same way as performed with SNMP Notifications.

NerveCenter version 5.0 supports only IPv4 ICMP monitoring and polling. IPv6 ICMP is not supported with respect to these abilities.

NerveCenter Support for ICMP Polling

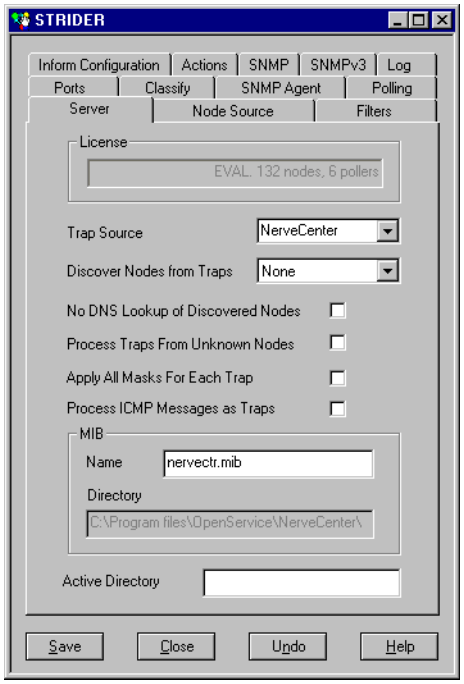
NerveCenter provides a means of accessing the ICMP “ping” operation. The manner of this interface is consistent with the product's SNMP access. A special Property “nl-ping”, defined by one of the MIB Modules provided with NerveCenter, is used to indicate that a poll is to perform an ICMP “ping”. The poll condition then handles examination of the results of the operation, much as NerveCenter handles SNMP polls. This mechanism includes special built-in triggers specific to ICMP polling. See *Chapter 8, Defining Property Groups and Properties* , *Chapter 9, Using Polls* and *Chapter 11, Using Other Data Sources* in *Designing and Managing Behavior Models*.

NerveCenter Support for ICMP Event Monitoring

NerveCenter can optionally be configured to treat incoming ICMP messages as events. This feature, available on UNIX and Linux, permits Trap Mask definitions to be created which filter ICMP alerts to the messages of interest. Use of the Trap Mask mechanism for this filtering makes the ability uniform to the event monitoring NerveCenter provides for SNMP. See *Using Trap Masks in Designing and Managing Behavior Models*.

Configuring NerveCenter for ICMP Event Monitoring

NerveCenter Administrator contains a **Process ICMP Messages as Traps** option on its Server tab. This control is used to enable/disable ICMP monitoring. ICMP monitoring is enabled only when this setting has been selected and the setting saved to the connected/managed NerveCenter Server. The parameter is persistent over restarts of the NerveCenter Server and does not require a restart of NerveCenter Server for a change to come into effect.


STRIDER

License: EVAL. 132 nodes, 6 pollers

Trap Source: NerveCenter

Discover Nodes from Traps: None

No DNS Lookup of Discovered Nodes ☐

Process Traps From Unknown Nodes ☐

Apply All Masks For Each Trap ☐

Process ICMP Messages as Traps ☐

MIB

Name: nervectr.mib

Directory: C:\Program files\OpenService\NerveCenter\

Active Directory:

When the control is enabled, all received ICMP messages are transformed into SNMPv1 Traps. Any applicable Trap Mask will then be applied to the event. The definition of Trigger Functions can be done in a manner entirely consistent with that used for SNMP event monitoring. The activation of triggers works identically. See *Using Trap Masks in Designing and Managing Behavior Models*.

When the control is disabled, NerveCenter Server continues to receive ICMP messages but the messages are processed as poll results (etc.) instead of being transformed into traps or forwarded to the user's trap masks.

Managing NerveCenter Security

11

NerveCenter security allows access for two different groups. A NerveCenter administrator must add users to an appropriate user group before they can access NerveCenter.

NOTE
NerveCenter can be installed without security on Windows, in which case users do not need to enter a name and password when logging on to the server.

On Windows, the NerveCenter Server authenticates users against the Windows built-in security. On UNIX, the NerveCenter Server authenticates users against the host's Pluggable Authentication Module (PAM) configuration. Membership in the appropriate groups determines what privileges users have within NerveCenter once they log in successfully.

NOTE
The User ID under which the NerveCenter Server runs must be a member of the NerveCenter Admins group.

Members of the NerveCenter Admins group can configure NerveCenter and create or modify behavior models. Members of the NerveCenter Users group can monitor network status and reset alarms. For more details about privileges, see the section *NerveCenter Login Rights on page 35*.

This chapter includes the following sections:

Section	Description
Managing Security on UNIX on page 166	Explains how to add users to user groups on UNIX.
Managing Security on Windows on page 170	Explains how to add users to user groups on Windows.
Troubleshooting: Managing NerveCenter Security on page 176	Provides troubleshooting tips related to security management.

Managing Security on UNIX

User logon attempts to a UNIX- or Linux-based NerveCenter Server are authenticated before access is permitted. Authentication checking occurs when the user attempts to log on with the NerveCenter Administrator or Client applications, the nccmd utility, or the NerveCenter Web Client interface. Authentication is performed whether the user is running the application on the same host as the NerveCenter Server or is accessing the NerveCenter Server remotely.

Beginning with version 5.0, NerveCenter integrates with a UNIX or Linux host’s Pluggable Authentication Module (PAM) subsystem to provide authentication. PAM allows NerveCenter logins to be based on the user accounts of the underlying operating system and subject to the authentication rules set up for that host.

To successfully access a NerveCenter Server, a login attempt must satisfy the following:

- The given username must match an existing user account
- The password must be valid for the given username
- The login attempt must be accepted by the host's PAM configuration as defined for the “nervecenter” service
- The account must be a member of either the NerveCenter Administrators (ncadmins) or NerveCenter Users (ncusers) group

NerveCenter does not require special treatment under PAM. The NerveCenter Server accesses PAM using the service name “nervecenter”, under which Administrators create specialized handling for NerveCenter access. If no specialized handling is in place, PAM defaults to matching NerveCenter authentication and account checking against the rules specified for the service name “other” or “OTHER”, depending on the rules by which the host's PAM subsystem operates. PAM is implemented and managed differently by each Linux and UNIX vendor; the network, host, or security administrator(s) for each NerveCenter Server will need to be aware of how PAM operates on their systems and configure it appropriately.

PAM configuration is required only on the Linux or UNIX system where the NerveCenter Server is installed. The NerveCenter Client, NerveCenter Administrator, and other NerveCenter components do not require PAM to be configured, as they use the NerveCenter Server for user account access functions. The user login functionality is handled only by the NerveCenter Server which a user is attempting to access, and the NerveCenter Server's PAM integration determines whether the login is successful.

NerveCenter's PAM integration provides administrators with more control over account access, allowing them to:

- Integrate account control with LDAP or NIS systems
- Select non-default password encryption methods
- Lock out user access with the /etc/nologin file
- Lock out users with expired passwords or blocked accounts

Such features are dependent upon the PAM implementation on the NerveCenter Server host. For example, Blowfish password encryption is supported on Sun Solaris 9 and 10, but not on RedHat Enterprise Linux.

The NerveCenter 5.0 installer for Linux and Solaris contains a PAM configuration section that briefly describes the host's PAM configuration with respect to NerveCenter. The installer registers NerveCenter with PAM under the service name “nervecenter”, through which PAM can provide enhanced authentication processing. (Examples of other service names are “passwd”, “login”, “rlogin” and “ssh”, which correspond to the Linux/UNIX applications of the same name; a service name does not need to match the application name.) NerveCenter’s PAM configuration controls access for any NerveCenter component attempting to log on to the NerveCenter Server. PAM authentication methods for the “nervecenter” service name are configured through a “nervecenter” file in the /etc/pam.d directory (Linux) or with “nervecenter” entries in the /etc/pam.conf file (Solaris). If no “nervecenter” configuration is provided, all systems authenticate using the rules for the default service name “other” or “OTHER”.

The installer places several sample configuration files in the /opt/OSInc/nc/install directory. Additionally, for Linux systems, a “nervecenter” file with a simple default setup is placed in the /etc/pam.d directory if such a directory exists and there is not already a “nervecenter” file present. For Solaris systems, no change is made to the pam.conf file; this is left to your system's administrators.

There is no user access to the PAM configuration from within NerveCenter. The NerveCenter Server accesses PAM to perform authentications and account checking but cannot otherwise alter, view, or reveal the PAM configuration. Users that log in to the NerveCenter Server cannot access their account, its definition, or alter aspects of it (changing an account password is not possible, for example). While PAM can handle different account- and access-related services, NerveCenter accesses the authentication and account management services only. The keywords PAM uses for these services are “auth” and “account” respectively. Thus the “nervecenter” configuration only needs to provide handling for these two services.

Troubleshooting UNIX Login Issues

NerveCenter Server reports the outcome of login attempts to syslog. NerveCenter Server events are sent to syslog as a member of the “daemon” facility and login attempt messages are sent with the Warning priority. The UNIX system administrator can use these values to help configure syslog with regards to NerveCenter messaging.

When a user is granted a login, syslog displays a message similar to

```
May 22 16:23:56 bluejay NerveCenter: [ID 651326 daemon.warning]
CATEGORY: 100  “nervectr” login granted with “ncadmins” access.
```

or

```
May 22 16:23:56 bluejay NerveCenter: [ID 651326 daemon.warning]
CATEGORY: 100  “nervectr” login granted with “ncusers” access.
```

When a user is denied a login, syslog displays a message similar to

```
May 22 16:23:04 bluejay NerveCenter: [ID 651326 daemon.warning]
CATEGORY: 100  login rejected for “nervectr”
```

To help diagnose the cause of denied logins, the UNIX NerveCenter Server can be started with a command-line switch that augments the above rejection message. To invoke this, (re)start NerveCenter Server using “ncstart -pamdebug”. Then monitor the syslog output as login attempts are made. The messages may arrive alongside messages from PAM modules which the “nervecenter” PAM configuration has invoked. The combination will assist your administrator diagnose the issue.

With the “-pamdebug” switch in effect, the following additions may be seen in syslog.

1. `... login rejected for “nervectr”: unknown user`

   However logins are defined on the UNIX host, the current setup does not include the user, “nervectr” in this example, as a defined account.

2. `... login rejected for “roger”: PAM authentication check fails`

   Here, the “roger” login matches to a defined account; however the PAM Authentication service has rejected the login attempt. This will correspond to the 'auth' entries within PAM for the “nervecenter” or else “other” configuration. In most cases, this will occur due to the submission of an incorrect password.

3. `... login rejected for “dharry”: PAM account management check fails`

   The “dharry” login attempt is rejected by the PAM Account Management service. This will correspond to the 'account' entries within PAM for the “nervecenter” or else “other” configuration. Rejections will be related to issues such a presence of a /etc/nologin file, a blocked account, an expired password and similar.

4. `“... login rejected for “tracy”: no group membership`

   The login attempt is denied because the user “tracy” is not a member of “ncadmins” or “ncusers”. However groups are defined on the UNIX host, the current setup does not include the user as required for NerveCenter Server access.

5. `“... login rejected for “matthew”: failure accessing groups`

   The login attempt is denied because the system's group definitions do not define “ncadmins” or “ncusers”.

Managing Security on Windows

Unless the person installing NerveCenter chose to install NerveCenter without security, NerveCenter Setup automatically creates two groups with the required login rights—NerveCenter Admins and NerveCenter Users. Setup also automatically adds to the NerveCenter Admins group the user ID of the person who installs NerveCenter.

NOTE
The person who installs NerveCenter must have the rights to create either local or global groups. If not, NerveCenter will not create these groups successfully.

Setup creates these groups differently depending on where you install NerveCenter.

- If you install on a Windows workstation or server, NerveCenter Setup creates two local groups. These two local groups must be on every machine that runs NerveCenter.
- If you install on an Windows primary or backup domain controller, NerveCenter Setup creates two global groups on the primary domain controller.

You can also create other groups that are members of one of the NerveCenter groups.

There are three scenarios for adding NerveCenter users:

- *Adding Users to a NerveCenter on a Windows Workstation or Server on page 170*
- *Adding Users to a NerveCenter on a Windows Domain Controller on page 172*
- *Working with Active Directory on page 174*

Adding Users to a NerveCenter on a Windows Workstation or Server

This section describes how to add users to NerveCenter groups if you install NerveCenter on a Windows workstation or server. Use the following guidelines when adding users to groups:

- Add NerveCenter administrators to the ncadmins group. See *The Role of a NerveCenter Administrator on page 36*.
- Add NerveCenter users to the ncusers group. NerveCenter users will only use the NerveCenter Client to monitor networks. See *NerveCenter Login Rights on page 35* for a more detailed discussion of permissions.

Remember, once you add users to a group, they must log out and back in for changes to take effect.

To add users to the local groups

- Use Windows User Manager to add your users to one of the two local groups to make sure they have the correct advanced user rights to use NerveCenter.

NOTE
You must have Admin rights on your local machine to add users to local groups.

To create global groups that are members of the local groups

1. On the primary domain controller, create one more global groups with the following advanced user rights:
   - Act as part of the operating system
   - Log on as batch
2. Use Windows User Manager for Domains to add your users to one of the two global groups.

   NOTE
   You must have Domain Admin rights to add users to the global groups.
3. Add these global groups as members of the NerveCenter Admins and NerveCenter Users groups on each local machine that is running a NerveCenter Server.

   Instead of global groups, you could add each of your users to the groups on each of the local machines manually. However, for convenience, we recommend using the global groups.

   NOTE
   You must have Admin rights on the local machine to add users to the local groups.

IF SOME OF YOUR USERS ARE ON ANOTHER DOMAIN, FOLLOW THESE ADDITIONAL STEPS

1. Add each user to one of the NerveCenter global groups on the domain to which the NerveCenter Server belongs.

   Doing this ensures that those users have the correct advanced user rights.

   NOTE
   You must have Domain Admin rights to add users to the global groups. You must also set up a two-way trust relationship between the two domains. See the Windows documentation for more information.

2. Tell your users to specify their domain along with their user IDs whenever NerveCenter prompts them for their user ID.

The NerveCenter Server authenticates users first by evaluating which group they belong to and then by evaluating the validity of their Windows user ID and password.

Adding Users to a NerveCenter on a Windows Domain Controller

This section describes how to set up the NerveCenter groups and membership in those groups if you install a NerveCenter Server on a Windows domain controller. Use the following guidelines when adding users to groups:

- Add NerveCenter administrators to the ncadmins group. See *The Role of a NerveCenter Administrator on page 36*.
- Add NerveCenter users to the ncusers group. NerveCenter users will only use the NerveCenter Client to monitor networks. See *NerveCenter Login Rights on page 35* for a more detailed discussion of permissions.

Remember, once you add users to a group, they must log out and back in for changes to take effect.

If you install the NerveCenter Server on a Windows domain controller, Setup creates the global groups NerveCenter Admins and NerveCenter Users automatically.

To Set Up Users

1. Use Windows User Manager for Domains to add your users to one of the two global groups.

   NOTE
   You must have Domain Admin rights to add users to the global groups.

2. Add these global groups as members of the NerveCenter Admins and NerveCenter Users groups on each local machine that is running a NerveCenter Server.

   NOTE
   You must have Admin rights on the local machine to add users to the local groups.

If Some of Your Users Are on Another Domain

1. Add each user to the NerveCenter global groups on the domain to which the NerveCenter Server belongs.

   Doing this ensures that those users have the correct advanced user rights.

   NOTE
   You must have Domain Admin rights to add users to global groups. You must set up a two-way trust relationship between the two domains. See your Windows documentation for more information.

2. Tell your users to specify their domains along with their user IDs whenever they are prompted for User ID by NerveCenter.

The NerveCenter Server authenticates users first by evaluating which group they belong to and then by evaluating the validity of their Windows user ID and password.

Working with Active Directory

If you are managing users and groups with Active Directory, NerveCenter can connect to your Active Directory server. NerveCenter still creates local NerveCenter Admins and NerveCenter Users groups, though you can manage user and administrator credentials through Active Directory.

Within Active Directory:

- Create NerveCenter Admins and NerveCenter Users groups.
- Add users to the appropriate groups.

After configuring NerveCenter to connect to the Active Directory server as described below, you can log in to NerveCenter using usernames and passwords defined with Active Directory.

CAUTION

Once you have connected to an Active Directory server, you can no longer use the local NerveCenter usernames and passwords; you can only log in with Active Directory credentials. Confirm that NerveCenter Admins Users exist on the Active Directory server before connecting NerveCenter to the Active Directory server.

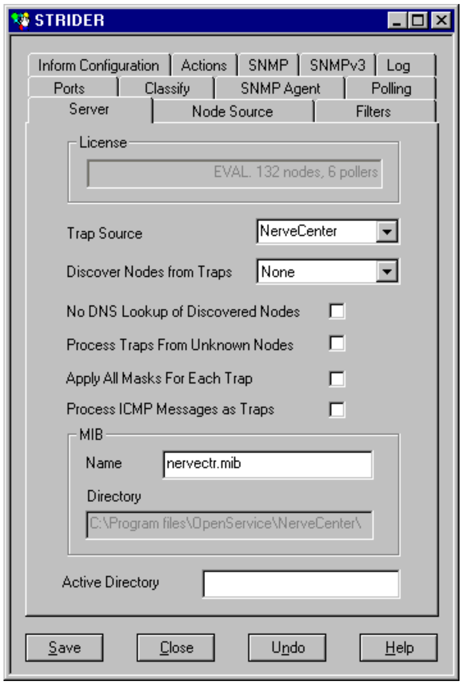
TO CONNECT TO AN ACTIVE DIRECTORY SERVER

1. Open NerveCenter Administrator and connect to the appropriate Server.

   See *Connecting Administrator to a NerveCenter Server on page 43*.

2. Select the **Server** tab.

   The Server page is displayed.


STRIDER

License: EVAL. 132 nodes, 6 pollers

Trap Source: NerveCenter

Discover Nodes from Traps: None

No DNS Lookup of Discovered Nodes ☐

Process Traps From Unknown Nodes ☐

Apply All Masks For Each Trap ☐

Process ICMP Messages as Traps ☐

MIB

Name: nervectr.mib

Directory: C:\Program files\OpenService\NerveCenter\

Active Directory:

3. Enter the full host name or IP address of the Active Directory Server in the Active Directory field.

   If you have configured Active Directory to use a port other than the default port of 389, you can also enter the port using the format *IPAddress*:*Port*.

4. Select **Save**.

Troubleshooting: Managing NerveCenter Security

The following list contains some common security problems NerveCenter users face. For UNIX-specific tips, see *Troubleshooting UNIX Login Issues on page 168*.

Users cannot connect to the server

Problem: Users are not members of the appropriate NerveCenter user group on the server machine (ncadmins or ncusers on UNIX or NerveCenter Admins or NerveCenter Users on Windows).

Solution: Check group membership. If users are not members of the appropriate groups, add them. Remember that on Windows, users must log out for changes to take place.

See *Managing Security on UNIX on page 166* or *Managing Security on Windows on page 170*.

Problem: Users do not have a valid Windows account.

Solution: Check the user account. See your Windows documentation for more details.

Problem: The server is not running.

Solution: Restart the server.

See *Chapter 4, Running the NerveCenter Server*

Users in ncadmins or ncusers on UNIX cannot connect to the server

Problem: ncadmins and ncusers have been created on a NerveCenter host that is also an NIS client.

Solution: When running NerveCenter on an NIS client, the groups ncadmins and ncusers must reside on the NIS Master. Local NerveCenter groups are ignored when the local system is an NIS client. Authentication is done on the NIS Master server. When running in an NIS environment, make sure the NerveCenter groups are defined on the NIS master. The user community members must be members of those groups as well.

See *Installing NerveCenter*.

Anyone can connect to a Windows server

Problem: NerveCenter was installed on Windows without security.

Solution: Reinstall the NerveCenter server with security.

See *Installing NerveCenter*.

User can monitor alarms, but can’t modify behavior models

Problem: Users who are members of ncusers (UNIX) or NerveCenter Users (Windows) can only monitor and reset alarms. To be able to modify NerveCenter objects, users must be members of ncadmins (UNIX) or NerveCenter Admins (Windows).

Solution: Add users to the appropriate groups.

See *Managing Security on UNIX on page 166* or *Managing Security on Windows on page 170*.

User cannot connect to the NerveCenter Web client.

Problem: User does not have the appropriate permissions on the Web server.

Solution: Give the user access to the Web server. See your Web server documentation for details.

11 Managing NerveCenter Security

Managing NerveCenter Alarm Actions

12

Alarm actions are important components of NerveCenter behavior models. The user supplies most of the data needed to perform the alarm action when the behavior model is created. However, before some alarm actions can be performed, an administrator must configure particular settings.

This chapter includes the following sections:

Section	Description
Specifying an SMTP Server for Mail Notification on page 180	Explains how to configure NerveCenter to send e-mail notifications via an SMTP server.
Configuring NerveCenter to Send Microsoft Mail on page 182	Explains the various scenarios in which you may configure NerveCenter to send messages via a Microsoft Exchange server.
Setting Dialing Properties for Paging Notification on Windows on page 188	Explains how to configure NerveCenter on Windows to send pages.
Setting Dialing Properties for Paging Notification on UNIX on page 189	Explains how to configure NerveCenter on UNIX to send pages.
Specifying Settings for Log Management on page 191	Explains how to configure NerveCenter to manage logs.
Troubleshooting: Managing Alarm Actions on page 194	Lists common problems NerveCenter users face when using alarm actions.

Specifying an SMTP Server for Mail Notification

In the event of noteworthy network conditions, NerveCenter can send email alerts. NerveCenter currently offers two options for having a behavior model send an e-mail message:

- With an SMTP server.
- With a Microsoft Exchange server. See *Configuring NerveCenter to Send Microsoft Mail on page 182*.

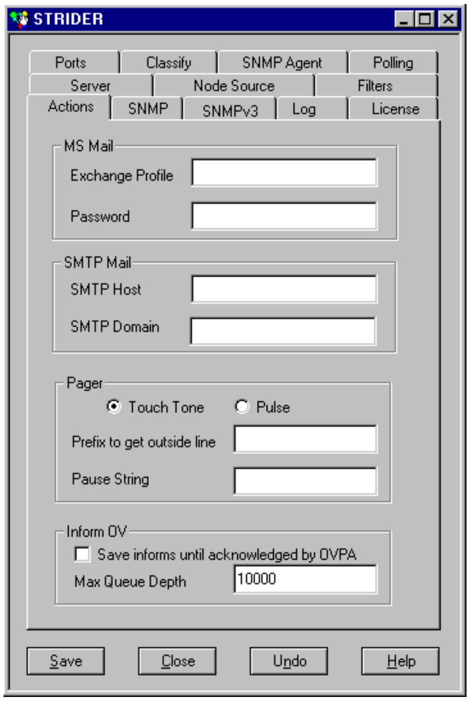
The recipient or recipients of the SMTP mail message is specified when the behavior model is created. However, before this message can be sent, the administrator must specify which SMTP server NerveCenter uses to send the message.

NOTE
On UNIX, the nervectr user is the sender of SMTP mail. To change this default sender, you can create an alias. Refer to your mail documentation for details.

TO SPECIFY A SMTP SERVER FOR MAIL NOTIFICATION

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server. See *Connecting Administrator to a NerveCenter Server on page 43*.
2. Select the **Actions** tab.

   The Actions tab is displayed.



Exchange Profile	
Password	
SMTP Host	
SMTP Domain	
[✓ Touch Tone]	Pulse
Prefix to get outside line	
Pause String	
☐ Save informs until acknowledged by OVPA	
Max Queue Depth	10000

3. In the **SMTP Host** field, type the name of the host running your SMTP mail server.
4. If you require a domain name other than the local host sending the mail, enter a valid domain name in the **SMTP Domain** field.

NOTE
If the **SMTP Domain** is empty, NerveCenter will use the local host name as the domain name.

5. Select **Save**.

Users can now include a SMTP Mail action in their behavior models.

Configuring NerveCenter to Send Microsoft Mail

In the event of noteworthy network conditions, NerveCenter can alert the necessary person via e-mail. NerveCenter currently offers two options for having a behavior model send an e-mail message:

- With an SMTP server. (See *Specifying an SMTP Server for Mail Notification on page 180*.)
- With a Microsoft Exchange server.

NOTE

To use the Microsoft mail alarm action, the NerveCenter Server must be running on a Windows platform.

The recipient or recipients of the Microsoft mail message is specified when the behavior model is created. However, before this message can be sent, the administrator must configure the Exchange profile that is sending the message.

There are two possible scenarios recommended for configuring the profile sending Microsoft Mail. In both cases a Microsoft Exchange Server administrator will need to create and designate a Microsoft Exchange mailbox specifically for NerveCenter purposes. The two different scenarios depend on how you will be running the NerveCenter Server:

- *Configuring NerveCenter to Send Microsoft Mail While Running as a Service on page 183*
- *Configuring NerveCenter to Send Microsoft Mail while Running the Server as a Process on page 186*

CAUTION

There are other scenarios in which the Microsoft Mail action can be used without creating a global NerveCenter group or account. However, they involve switching Exchange profiles each time a new user logs in to the NerveCenter machine.

Configuring NerveCenter to Send Microsoft Mail While Running as a Service

It is recommended that you run the NerveCenter Server as a Windows service because is simplifies accounts and other network domain concerns. For NerveCenter to send Microsoft mail properly, a network administrator must create a separate NerveCenter account and Exchange mailbox.

TO CONFIGURE NERVECENTER TO SEND MICROSOFT MAIL WHILE RUNNING AS A SERVICE

1. Have your Microsoft Exchange Server administrator create a NerveCenter mailbox.

   Although the mailbox name could be anything, for ease of use it is recommended you ask your Microsoft Exchange Server administrator to use the name NerveCenter.

2. Have your network administrator create an account in the Primary Domain Controller with access rights to the NerveCenter-designated mailbox created in step 1.

   Although the account name could be anything, for ease of use it is recommended you ask your administrator to use the name NerveCenter.

3. Change the account under which the NerveCenter Server service will be running.

   By default, when NerveCenter Server is installed as a service it is placed under the system account. To change the account:

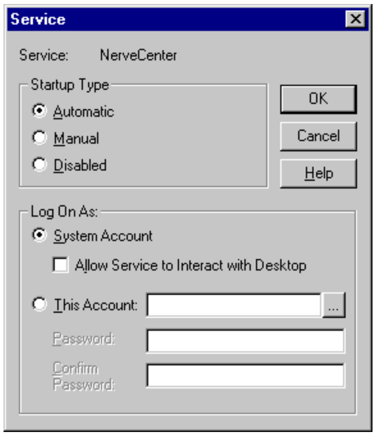
   a. From the **Start** menu, select **Settings**, then **Control Panel**.

   b. Double-click the **Services** icon.

      The Services window appears.

   c. In the Service list, highlight NerveCenter and select **Startup**.

      The NerveCenter Service window opens.


Service

Service: NerveCenter

Startup Type
[✓ Automatic] Manual Disabled

Log On As:
[✓ System Account]
☐ Allow Service to Interact with Desktop
This Account:
Password:
Confirm Password:

d. In the **Log On As** area, select **This Account**.

   The System Account setting is deselected.

e. In the **This Account** field, select the NerveCenter-designated account created in step 2.

f. In the **Password** and **Confirm Password** fields, type the password assigned to the NerveCenter-designated account.

g. Select **OK**.

You have now specified that the NerveCenter Server service will run under the NerveCenter-designated account that has access rights to the NerveCenter-designated mailbox in the Microsoft Exchange Server.

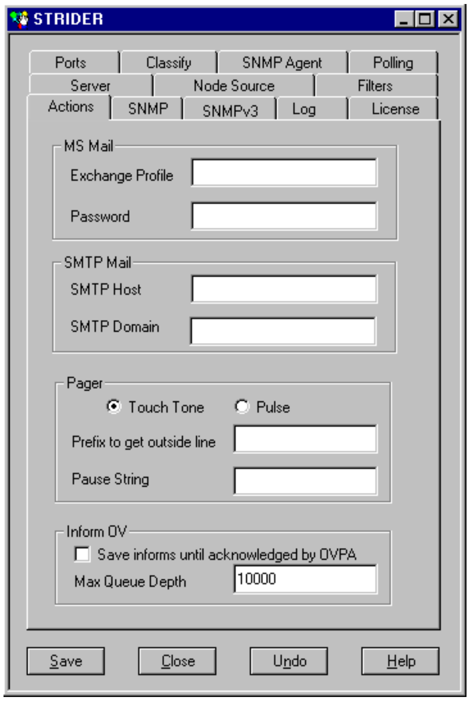
4. On the machine that will run NerveCenter Server, create a NerveCenter-designated Exchange profile to point to the machine hosting the Microsoft Exchange Server and the NerveCenter-designated mailbox created in step 1.

   Although the account name could be anything, for ease of use it is recommended you use the name NerveCenter.

5. Open the NerveCenter Administrator and connect to the appropriate NerveCenter Server. See *Connecting Administrator to a NerveCenter Server on page 43*.

6. Select the **Actions** tab.

   The Actions tab appears.



MS Mail

Exchange Profile	
Password	

SMTP Mail

SMTP Host	
SMTP Domain	

Pager

[✓ Touch Tone] Pulse

Prefix to get outside line	
Pause String	

Inform OV

☐ Save informs until acknowledged by OVPA

Max Queue Depth	10000

7. In the **Exchange Profile** field, type the name of the NerveCenter-designated profile created in step 4.
8. Select **Save**.

Users can now include a Microsoft Mail action in their behavior models. The message will be sent from the mailbox created in step 1.

Configuring NerveCenter to Send Microsoft Mail while Running the Server as a Process

It is recommended that you run the NerveCenter Server as a Windows service because is simplifies accounts and other network domain concerns. See *Configuring NerveCenter to Send Microsoft Mail While Running as a Service on page 183*.

Should you choose, however, to run NerveCenter Server as a process, you must have a network administrator create a global group for all your NerveCenter administrators as well as a Microsoft Exchange mailbox specifically for NerveCenter.

TO CONFIGURE NERVECENTER TO SEND MICROSOFT MAIL WHILE RUNNING THE NERVECENTER SERVER AS AN APPLICATION BY ESTABLISHING A GLOBAL GROUP

1. Have your network administrator create a global group in the Primary Domain Controller. Include in this group any user who should have administrator privileges in NerveCenter.

   Although the account name could be anything, for ease of use it is recommended you ask your network administrator to use the name NerveCenter Admins.

2. Have your Microsoft Exchange Server administrator create a mailbox for NerveCenter purposes.

   Although the mailbox name could be anything, for ease of use it is recommended you ask your Microsoft Exchange Server administrator to use the name NerveCenter.

   NOTE
   The Exchange Server administrator should make sure the Primary Windows Account for this NerveCenter-designated mailbox is the NerveCenter administrators-designated global group created in step 1.

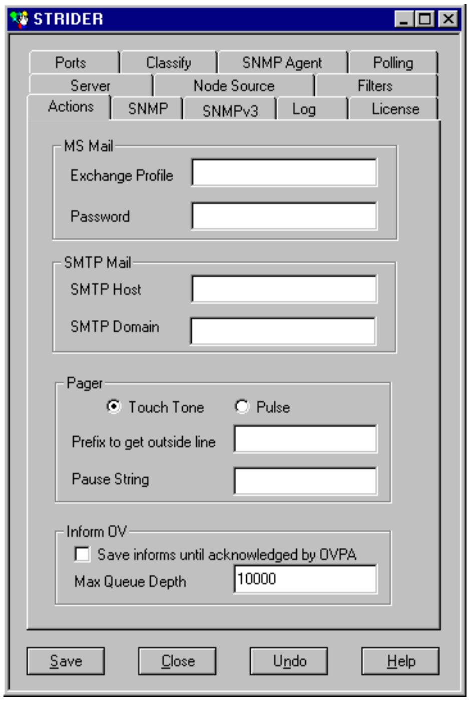
3. On each machine that will run NerveCenter Server, create a NerveCenter-designated Exchange profile to point to the machine hosting the Microsoft Exchange Server and the NerveCenter-designated mailbox created in step 1.

   Although the profile name could be anything, for ease of use it is recommended you use the name NerveCenter.

4. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server. See *Connecting Administrator to a NerveCenter Server on page 43*.

5. Select the **Actions** tab.

   The Actions tab appears.



STRIDER

MS Mail

Exchange Profile

Password

SMTP Mail

SMTP Host

SMTP Domain

Pager

● Touch Tone ○ Pulse

Prefix to get outside line

Pause String

Inform OV

☐ Save informs until acknowledged by OVPA

Max Queue Depth 10000

6. In the **Exchange Profile** field, type the name of the NerveCenter-designated profile created in step 4.

7. Select **Save**.

8. All new users assigned administrator privileges locally to a NerveCenter Admins group must be added to the global NerveCenter Administrators-designated group as well.

   Users can now include a Microsoft Mail action in their behavior models. The message will be sent from the mailbox created in step 2.

Setting Dialing Properties for Paging Notification on Windows

In the event of noteworthy network conditions, NerveCenter can send a page. The recipient's pager data is entered when the behavior model is created, though an administrator must configure the server's dialing properties before a page can be sent.

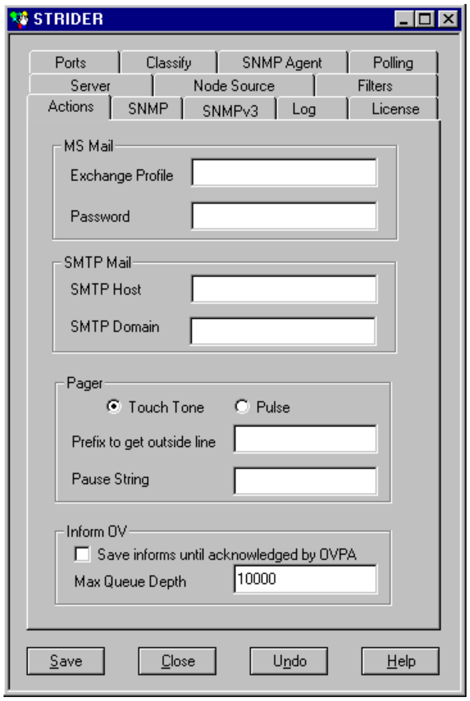
This section describes how to set dialing properties for a NerveCenter Server on Windows. See *Setting Dialing Properties for Paging Notification on UNIX on page 189*.

NOTE
The paging server must have access to a TAPI-compliant modem and outside line.

To configure NerveCenter to perform the paging action on Windows

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server. See *Connecting Administrator to a NerveCenter Server on page 43*.
2. Select the **Actions** tab.

   The Actions tab is displayed.



MS Mail
Exchange Profile
Password

SMTP Mail
SMTP Host
SMTP Domain

Pager
[✓ Touch Tone] Pulse
Prefix to get outside line
Pause String

Inform OV
☐ Save informs until acknowledged by OVPA
Max Queue Depth 10000

3. In the **Pager** area, select either **Touch Tone** or **Pulse**.
4. In the **Prefix to get outside line** field, type the number or numbers for an outside line.

   In many offices, for example, you dial 9 for an outside line.
5. In the **Pause String** field, type the characters used to create a pause between dialing and sending data.
6. Select **Save**.

Users can now include a Paging action in their behavior models.

Setting Dialing Properties for Paging Notification on UNIX

In the event of noteworthy network conditions, NerveCenter can send a page. The recipient's pager data is entered when the behavior model is created, though an administrator must configure the server's dialing properties before a page can be sent.

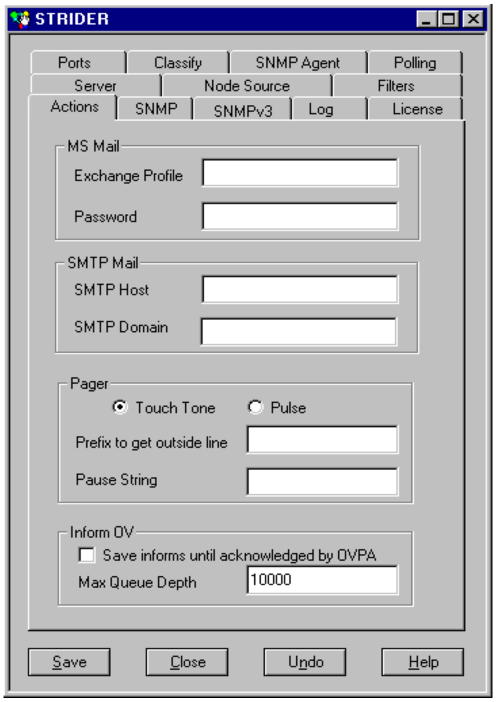
This section describes how to set dialing properties for a NerveCenter Server on UNIX. See also *Setting Dialing Properties for Paging Notification on Windows on page 188*.

NOTE
The paging server must be on a machine with a modem and access to a outside line.

TO CONFIGURE NERVECENTER TO PERFORM THE PAGING ACTION ON UNIX

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server. See *Connecting Administrator to a NerveCenter Server on page 43*.
2. Select the **Actions** tab.

   The Actions tab is displayed.



Pager: [✓ Touch Tone] Pulse

[ ] Save informs until acknowledged by OVPA

Max Queue Depth: 10000

3. In the **Log File** field, type the full path and name of the pager log file.

   The Pager action logs success or failure messages to this file, which contain the following:

   - a date and time stamp
   - the process ID (PID) of the Pager process
   - the log message itself

   If the log file cannot be opened, all log messages are directed to standard error (stderr).

4. In the **Maximum Retries** field, type the maximum number of times the system should try to dial the pager number.
5. In the **Retry Interval** field, type the number of seconds to pause between dialing attempts.
6. In the **Modem Speed** field, type the speed of the modem used for paging.
7. In the **Pager Port** field, type the location of the port used for paging.
8. Select **Save**.

Users can now include a Paging action in their behavior models.

Specifying Settings for Log Management

NerveCenter offers two ways that a behavior model can log network data:

- The Log to File alarm action writes information about an alarm transition to an ASCII text file.
- The Log to Database alarm action, available only on Windows systems, writes information about an alarm transition to the NerveCenter database. You can extract logged data from the database using any ODBC-compliant reporting tool.

When a behavior model is created, the user specifies the name of the log as well as the type of data that will be recorded. However, an administrator must specify how NerveCenter will manage these logs.

NerveCenter logs have changed over the course of several releases. Some of the more significant characteristics of NerveCenter logs include:

- Time fields appear in the following format:

  ```
  mm/dd/yy hh.mm.ss day
  ```

  For example, 10/04/98 12.03.44 Wed
- NerveCenter will delimit fields with a semi-colon
- No spaces will appear around the equal sign (=)
- Values will appear in the following format:

  ```
  attribute.instance=value
  ```

  For example, ifInOctets.3=5
- NerveCenter will print all fields when a default format is chosen. For example, the same trap in verbose and nonverbose format:

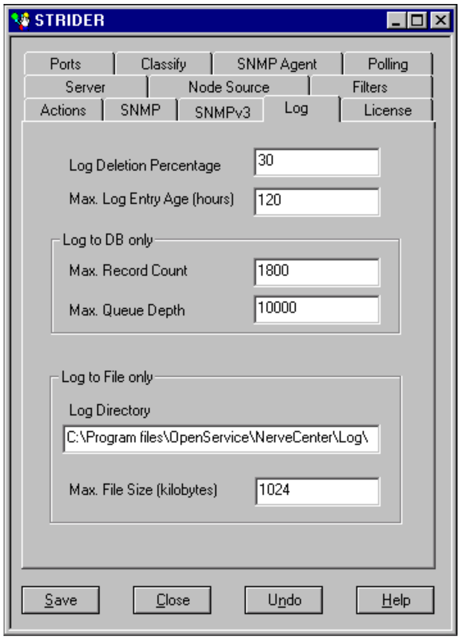
  ```
  verbose: Time=11/11/2002 15:14:43 Mon; LogId=4894;
  DestStateSev=Normal; NodePropertyGroup=Mib-II;
  NodeName=MyComputer; AlarmName=AllTraps_LogToFile;
  OrigState=Ground; TriggerName=allTraps; DestState=Logging;
  TrapPduTime=321; TrapPduGenericNumber=4;
  TrapPduEnterprise=1.3.6.1.41.78; TrapPduSpecificNumber=0;
  TriggerInstance=; TriggerBaseObject=

  nonverbose: 11/11/2002 15:14:43 Mon;4875;Normal;Mib-
  II;MyComputer;AllTraps_LogToFile;Logging;allTraps;Logging;321;4;1.3.
  6.1.41.78;0;;
  ```

To specify the settings NerveCenter uses to manage logging

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server. See *Connecting Administrator to a NerveCenter Server on page 43*.
2. Select the **Log** tab.

   NerveCenter displays the Log tab.



Log Deletion Percentage: 30
Max. Log Entry Age (hours): 120

Log to DB only
Max. Record Count: 1800
Max. Queue Depth: 10000

Log to File only
Log Directory: C:\Program files\OpenService\NerveCenter\Log\
Max. File Size (kilobytes): 1024

3. In the **Log Directory** field, type the complete path of the directory where NerveCenter stores the log.

   By default, this field specifies a Log folder created in the NerveCenter directory during installation.
4. In the **Max Log Entry Age** field, type the number of hours you want to keep individual log entries before they are deleted from the rest of the log.

   The way NerveCenter uses this number depends on which logging action is performed:

   - Log to File: NerveCenter checks every file every 20th entry. It deletes any entry it finds that is older than the maximum age specified.
   - Log to Database: NerveCenter waits until the log reaches its size limit. It then deletes any entry it finds that is older than the maximum age specified.

5. In the **Max Log File Size** field, type the size limit, in kilobytes, of the ASCII text file storing the results of the Log to File alarm action.

   This field does not affect the Log to Database alarm action.

6. In the **Max. Number of Records** field, type the highest number of records in the database file storing the results of the Log to Database alarm action.

   This field does not affect the Log to File alarm action.

7. In the **Max. Queue Depth** field, type the highest number of changes that you want queued before saving to the database.

   When the maximum is reached, no more changes are saved until the queue is reduced. You can have more than one change queued for a single database record.

8. In the **Log Deletion Percentage** field, type the percentage of the log to clear when the maximum file size or the maximum number of records is reached.

   This field applies to both the Log to File and Log to Database alarm actions:

   - Log to File: NerveCenter checks a log file every 20th entry. If the file log exceeds the amount specified in the Max Log File Size, NerveCenter deletes the percentage specified here, starting with the oldest entries.
   - Log to Database: NerveCenter checks a database log every 20th entry. If the database log exceeds the amount specified in the Max. Number of Records field, NerveCenter first deletes any records older than the age specified by the Max Log Entry Age. If the database log still exceeds the allowable amount, NerveCenter removes the percentage specified here, starting with the oldest entries.

9. Select **Save**.

NerveCenter will manage logs according to these settings whenever the Log to File or Log to Database alarm actions are used in a behavior model.

Troubleshooting: Managing Alarm Actions

The following list contains some common problems users have when using alarm actions.

NerveCenter is not sending SMTP mail notifications

Problem: A SMTP Server is not specified.

Solution: Specify a SMTP host under the Actions tab of the NerveCenter Administrator.

See *Specifying an SMTP Server for Mail Notification on page 180*.

Microsoft mail notifications are not being sent as NerveCenter

Problem: An incorrect Microsoft Exchange profile is specified.

Solution: Create a correct Microsoft Exchange profile and specify the profile under the Actions tab of the NerveCenter Administrator.

See *Configuring NerveCenter to Send Microsoft Mail on page 182*.

NerveCenter is not dialing pages correctly

Problem: The dialing properties are incorrect.

Solution: Set the correct dialing properties for paging under the Actions tab of the NerveCenter Administrator.

See *Setting Dialing Properties for Paging Notification on Windows on page 188* or *Setting Dialing Properties for Paging Notification on UNIX on page 189*.

The logs created by my behavior models are using too much disk space

Problem: The size limits for NerveCenter logging actions are too high.

Solution: Set the maximum sizes to an adequate limit.

See *Specifying Settings for Log Management on page 191*.

Managing NerveCenter Web Integration

13

NerveCenter alarms can be monitored using a standard web browser. To connect to a NerveCenter Server, the NerveCenter Web Collector must be installed and running on the machine hosting your web server. As an administrator, you must make sure the Web Collector is running and your web server is configured to work with NerveCenter.

This chapter includes the following section:

Section	Description
How to Start and Stop the NerveCenter Web Collector on page 196	Explains how to start and stop the NerveCenter Web Collector to allow for web integration.
Configuring a Web Server for NerveCenter on page 197	Explains how to configure a web server if it was not present when NerveCenter was installed.

How to Start and Stop the NerveCenter Web Collector

During a typical installation, the NerveCenter Web Collector is installed as a Windows service or UNIX daemon. Therefore, whenever you boot the host machine, the NerveCenter Web Collector will automatically start.

There may be times when you want to start or stop the NerveCenter Web Collector manually. The procedure depends on your operating system:

- *Starting and Stopping the NerveCenter Web Collector in UNIX on page 196*.
- *Starting and Stopping the NerveCenter Web Collector in Windows on page 196*.

Starting and Stopping the NerveCenter Web Collector in UNIX

During a typical installation, the NerveCenter Web Collector is installed as a daemon so it automatically starts whenever the host machine is booted. However, there may be times when you want to start or stop the NerveCenter Web Collector manually.

TO START OR STOP THE NERVECENTER WEB COLLECTOR

- To start the NerveCenter Web Collector, type at the command line:

  ```
  ncwebstart
  ```

- To stop the NerveCenter Web Collector, type at the command line:

  ```
  ncwebstop
  ```

Starting and Stopping the NerveCenter Web Collector in Windows

During a typical installation, the NerveCenter Web Collector is installed as a Windows service. Therefore, the NerveCenter Web Collector automatically starts whenever the host machine boots.

To stop and start the NerveCenter Web Collector, you must use the Services Applet found in the Windows Control Panel. See your Windows documentation for more details on starting and stopping services.

Configuring a Web Server for NerveCenter

If NerveCenter web support was installed on a machine with a supported web server installed and running, NerveCenter Setup (or the installation script on UNIX) already configured your web server. Users should be able to connect immediately.

If NerveCenter web support was installed on a machine without an installed and running web server, you must configure your web server by doing the following:

1. Set up a NerveCenter directory that links to the Web subdirectory in the NerveCenter installation directory and that has read access.
2. Set up a NerveCenterCGI directory that links to the Web\CGI subdirectory in the NerveCenter installation directory and that has read and execute access.
3. Make sure that either index.html or default.htm are supported as default documents.

The following sections describe how to configure web servers to for the NerveCenter web client:

- *Configuring Internet Information Server*
- *Configuring an Apache Web Server*

Configuring Internet Information Server

The following sample procedure illustrates how to configure Internet Information Server (IIS) on Windows to support NerveCenter.

NOTE
See your web server documentation for complete instructions.

TO CONFIGURE INTERNET INFORMATION SERVER ON WINDOWS NT

1. Start the Internet Service Manager.
2. Select **Service Properties** for the server you chose when you installed NerveCenter.

   NOTE
   Any web server NerveCenter uses must use the default port 80.

3. Select the **Directories** tab.
4. Create the NerveCenter directory that links to the Web subdirectory by doing the following:
   a. Select **Add**.
   b. In the **Directory** field, type the full path of the Web subdirectory.

      For example, if you installed in the default location, the path would be C:\Program Files\OpenService\NerveCenter\Web.

   c. In the **Alias** field, type NerveCenter.
   d. On the **Access** panel, select the **Read** check box.
   e. Select **OK**.
5. Create the NerveCenterCGI directory that links to the Web\CGI subdirectory:
   a. Select **Add**.
   b. In the **Directory** field, type the full path of the Web subdirectory.

      For example, if you installed in the default location, the path would be C:\Program Files\OpenService\NerveCenter\Web\CGI.

   c. In the **Alias** field, type NerveCenter**CGI**.
   d. On the **Access** panel, select the **Read** and **Execute** check boxes.
   e. Select **OK**.

6. Check the **Enable Default Document** check box.
7. If the **Default Document(s)** field doesn't already include **index.html** and **default.htm**, add them. Separate multiple entries in this field with commas.
8. If Web publishing is off, turn on the web server.

On Windows 2000 and XP machines, IIS should be configured automatically when you select Web Support during installation. However, you should confirm that the settings are correct.

To confirm IIS is configured on Windows 2000 and XP

1. Open the Internet Information Services dialog box by selecting **Start** > **Settings** > **Control Panel** > **Administration Tools** > **Internet Service Manager**.
2. On the **Tree** tab, expand the name of the server on which you installed NerveCenter Server.
3. Expand **Default Web Site** and select NerveCenter.
4. Select **Action** > **Properties**.

   The NerveCenter Properties dialog box opens.
5. Confirm the following on the **Virtual Directory** tab:
   a. The Local Path is C:\Program Files\OpenService\NerveCenter\Web, if you installed in the default location
   b. The **Read** check box is selected.
   c. **Scripts and Executables** is selected from the Execute Permissions list.
6. Click **OK** to close the NerveCenter Properties dialog box.
7. Repeat steps 3 and 4 this time selecting NerveCenter**CGI**.
8. Confirm the following on the **Virtual Directory** tab:
   a. The Local Path is C:\Program Files\OpenService\NerveCenter\Web\CGI, if you installed in the default location
   b. The **Read** check box is selected.
   c. **Scripts and Executables** is selected from the Execute Permissions list.
9. Close the Internet Information Services dialog box.

If the NerveCenter and NerveCenterCGI directories were not created during installation, you can create them using the Virtual Directory Creation Wizard.

To create the NerveCenter and NerveCenterCGI directories on Windows

1. From the Internet Information Services dialog box, select **Default Web Site**.

   For instructions on accessing the Internet Information Services dialog box, see *To confirm IIS is configured on Windows 2000 and XP on page 199*.

2. Select **Action** > **New** > **Virtual Directory**.

   The Virtual Directory Creation Wizard opens.

3. Click **Next**.

   The Virtual Directory Alias dialog box appears.

4. In the **Alias** field, enter NerveCenter and click **Next**.

   The Web Site Content Directory dialog box appears.

5. In the **Directory** field, enter NerveCenter*InstallationPath*\Web and click **Next**.

   The Access Permission dialog box appears.

6. Select the **Read** and **Execute** checkboxes and click **Next**.

7. Click **Finish**.

8. Repeat step 2 through step 7 to create a virtual directory named NerveCenterCGI with the Web Site Content Directory NerveCenter*InstallationPath*\Web\\CGI.

   You may also need to create virtual directories for the subdirectories in NerveCenter*InstallationPath*\Web (**Conf** and **Help**)

NOTE
If you have another web server running (FastTrack, for example), you must shut down the other web server before you can start IIS.

Configuring an Apache Web Server

To configure an Apache web server, you must edit the **httpd.conf** file as follows.

To configure an Apache web server

1. Edit the file to use port 80.
2. Add the following lines (for UNIX, change C:\Program Files\OpenService\NerveCenter to /opt/OSInc):

```
Alias /NerveCenter/ “C:/Program Files/OpenService/NerveCenter/web/”

<Directory “C:/Program Files/OpenService/NerveCenter/web”>
   Options Indexes FollowSymLinks MultiViews IncludesNoExec
   AddOutputFilter Includes html
   AllowOverride None
   Order allow,deny
   Allow from all
</Directory>

ScriptAlias /NerveCenterCGI/
“C:/ProgramFiles/OpenService/NerveCenter/web/cgi/”

<Directory “C:/Program Files/OpenService/NerveCenter/web/cgi”>
   AllowOverride None
   Options None
   Order allow,deny
   Allow from all
</Directory>
```

By entering **http://server/**NerveCenter in a standard web browser, where server is the name of the machine on which your web server is running, users can now monitor a network.

13 Managing NerveCenter Web Integration

Managing the Database

14

The NerveCenter database does not require much maintenance, whether your database is a Microsoft SQL Server or Microsoft Access database on Windows or flat files on UNIX. You can view information about the database as well as statistics about the information it contains, back up and restore the database, and transfer the contents of one database to another.

NOTE
To install a new database, see *Installing NerveCenter*. You must be a Windows administrator or UNIX root to run SerializeDB.

This chapter includes the following sections:

Section	Description
Database Formats on page 204	Describes the NerveCenter database formats on each platform.
Viewing Information about the Database on page 205	Describes how to view information about the database, including the name of the data source, the name of the database, whether it is connected, and the number of each type of object it contains.
NerveCenter Database Utilities on page 206	Describes the NerveCenter database utilities SerializeDB and DBWizard.
Backing up the Database on page 209	Describes when and how to back up the contents of the database, using the SerializeDB application or running SerializeDB from the command line.
Restoring the Database on page 213	Describes when and how to restore the contents of the database, using the SerializeDB application or running SerializeDB from the command line.
Transferring Data between Databases on page 218	Describes how to transfer the contents of the database to another database—even a database on another platform or of a different type.
ncdb2html.pl on page 219	Describes how to use the ncdb2html.pl script to convert the NerveCenter database to HTML.
Troubleshooting: Managing the NerveCenter Database on page 224	Lists some of the common problems users face when working with the NerveCenter database.

Database Formats

NerveCenter supports three different database formats—use the most appropriate one for your environment. The three database formats are:

- Microsoft SQL Server (Windows)

  You determine the Microsoft SQL Server database name when you install NerveCenter and run DBWizard to create the database. The database is located on the SQL server machine and the ODBC connection string on the NerveCenter server machine points to the database.

- Microsoft Access (Windows)

  The name of the Microsoft Access database file is **NCAccess.mdb**. The file must be located on the server machine. (You do not have to have Microsoft Access installed on the server machine.)

- Flat files (UNIX)

  The files that make up the UNIX database are **NerveCenter.ncdb**, which contains information about the NerveCenter objects such as polls and alarms, and NerveCenter.node, which contains information about nodes.

NOTE

In the past, the UNIX database was made up of the following files: ***.ncdb**, **alarm_sum.def**, **sysobjid.dat**, ***.ovhosts**, and ***.ncdb.nts**.

When NerveCenter was installed, the installer set up one of these databases. If another format or platform would be more appropriate, you can create a new database on that platform (as long as you have a server on that platform) and transfer your data from the old database to the new one. See *Installing NerveCenter* for instructions on creating a new database. Also see *Transferring Data between Databases on page 218* for instructions on transferring data from one database to another.

Viewing Information about the Database

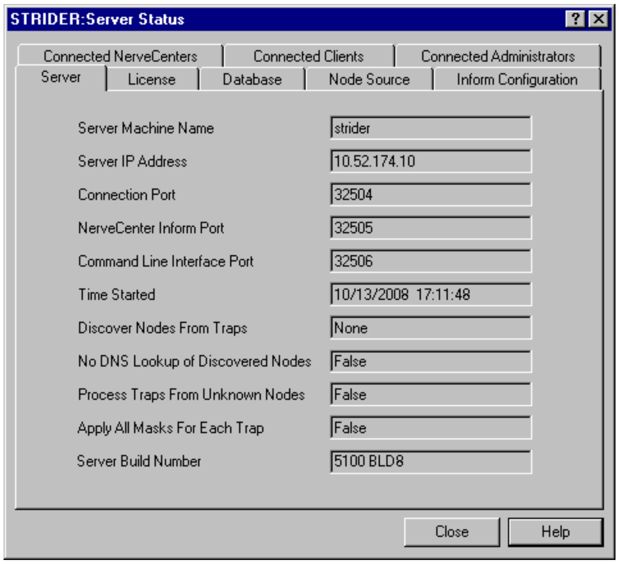
From the NerveCenter Client or Administrator, you can find out the name of the data source, the machine on which the database is kept (if it is different from the machine on which the server runs), the name and location of the database, and whether the database is connected. You can also view the number of alarms, polls, nodes, masks, property groups, and properties stored in the database.

Information about the database is useful when you are troubleshooting or when you just want to verify the database information—for example, to make sure you are going to back up the contents of the correct database.

TO VIEW INFORMATION ABOUT THE DATABASE

1. If you are connected to more than one server from either the administrator or client, make sure the server for which you want the database information is the active server.
2. From the **Server** menu, select **Server Status**.

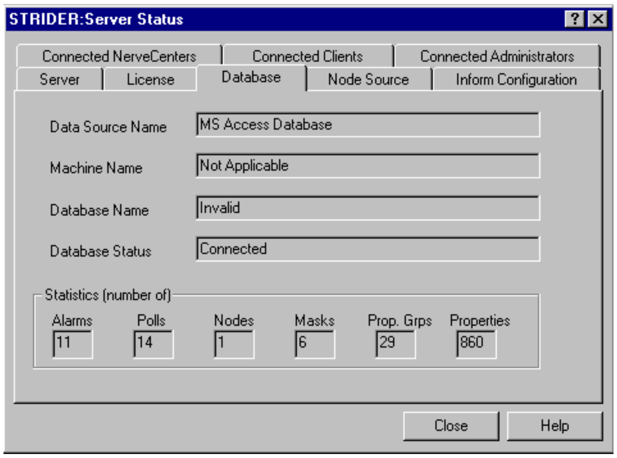
   The Server Status window is displayed.



Server Machine Name	strider
Server IP Address	10.52.174.10
Connection Port	32504
NerveCenter Inform Port	32505
Command Line Interface Port	32506
Time Started	10/13/2008 17:11:48
Discover Nodes From Traps	None
No DNS Lookup of Discovered Nodes	False
Process Traps From Unknown Nodes	False
Apply All Masks For Each Trap	False
Server Build Number	5100 BLD8

3. Select the **Database** tab.

   The Database tab is displayed.



STRIDER:Server Status

Data Source Name: MS Access Database

Machine Name: Not Applicable

Database Name: Invalid

Database Status: Connected

Statistics (number of)

Alarms	Polls	Nodes	Masks	Prop. Grps	Properties
11	14	1	6	29	860

4. When you are done, select **Close** to close the Server Status window.

After you view the database information, you will know what database NerveCenter is using, whether the database is connected, and how many objects of each type are stored in the database.

NerveCenter Database Utilities

NerveCenter includes two utilities that facilitate data management.

Utility	Description
SerializeDB	A conversion utility that transfers NerveCenter data to and from a serialized file.
DBWizard	A Windows iinstallation and upgrade utility. DBWizard works with SerializeDB and InstallDB to install, upgrade, or connect to a Microsoft Access or SQL Server database.

NerveCenter uses different methods of storing information on UNIX and Windows systems. The following sections describe data management for each platform:

- *UNIX on page 207*
- *Windows on page 208*

UNIX

NerveCenter uses two files for data storage on UNIX:

- **NerveCenter.ncdb**—Contains information about NerveCenter objects, such as polls, masks, and alarms.
- **NerveCenter.node**—Contains information about the nodes that NerveCenter manages.

Both files are loaded into NerveCenter during installation. After installation, you can back up, restore, and transfer this data by using SerializeDB. *Figure 14-1* shows the UNIX database files.

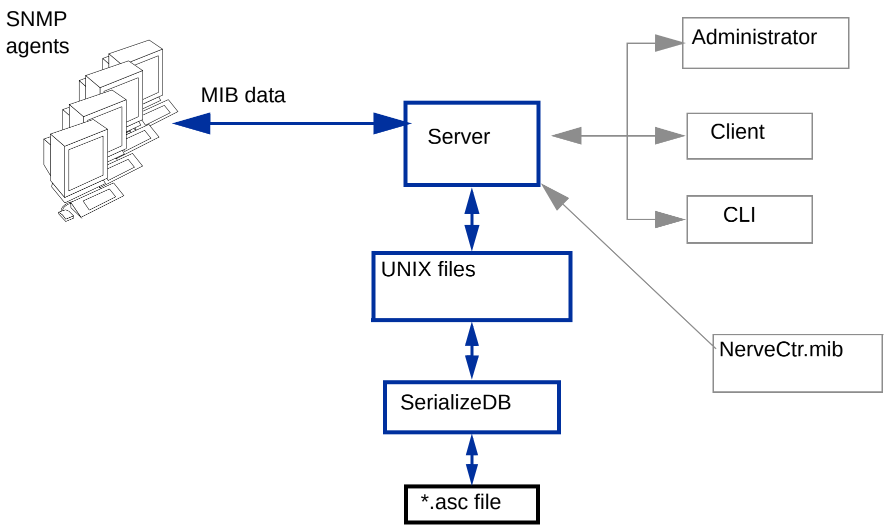


FIGURE 14-1. NerveCenter Data Storage Files on UNIX

SerializeDB performs two operations:

- Converts UNIX *.ncdb and *.node files to a single database file, whose name you provide at the time of conversion. The database file has a binary format with an .asc extension.
- Converts the same or another *.asc file to two *.ncdb and *.node files.

You can use the serialized file as a backup. Should you need to restore the backup, convert the file back to your original *.ncdb and *.node files. You can also transfer the serialized file to another NerveCenter machine and convert the file to *.ncdb and *.node files for that NerveCenter. Finally, you can transfer your original database files to a Windows NerveCenter and import the files.
For more information about multiple NerveCenters, see *Using Multiple NerveCenter Servers on page 31*.

Windows

On Windows, NerveCenter stores behavior model objects in a database that is accessed using ODBC. *Figure 14-2* shows the Windows database files with the other NerveCenter components.

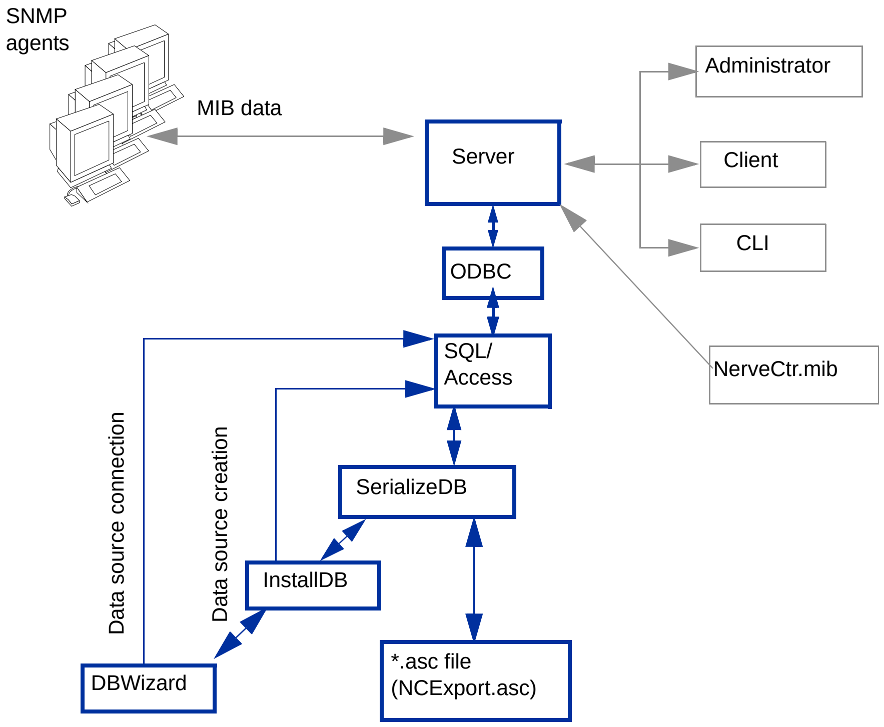


FIGURE 14-2. NerveCenter Database Components on Windows

NerveCenter ships with a serialized file that is imported into the database during installation. The file, NCExport.asc, contains the base set of NerveCenter objects, such as polls, masks, and alarms. During installation, DBWizard allows you to enter settings, for example, an ODBC source and a SQL Server host. DBWizard invokes an installation tool called InstallDB to configure the SQL Server or Access database. InstallDB, in turn, invokes SerializeDB to convert NCExport.asc to the proper database format. InstallDB can also be run standalone with SerializeDB to perform unattended database installations.

Afterward, if you upgrade to a new version of NerveCenter, DBWizard can be used to run an upgrade script against your database. Upgrade requirements for each new release are included in the *NerveCenter Release Notes*.

As with the UNIX environment, you can use SerializeDB to transfer data between your NerveCenter database and a serialized file. This lets you back up, restore, and transfer a database. On Windows, SerializeDB performs the following operations:

- Converts your database to a single database file. The database file has a binary format with an .asc extension.
- Converts an *.asc file to a database.
- Converts your database to NerveCenter flat files (*.ncdb and *.node), which can then be imported to another NerveCenter on a UNIX machine.
- Converts UNIX *.ncdb and *.node files to a SQL Server or Access database.

For information about working with multiple NerveCenters, see *Using Multiple NerveCenter Servers on page 31*. You can also use DBWizard as an interface that runs SerializeDB and InstallDB to import a serialized file that you specify in DBWizard.

Backing up the Database

You should establish a regular database backup schedule that suits your environment. Also, you should back up your database prior to doing anything that might affect the database. For example, you'll want to back up the database before upgrading NerveCenter or before importing data from another database. Then, if you need to restore the original data, you can.

SerializeDB exports the object and node data from the database to a single serialized ASCII file (*.asc). On Windows, SerializeDB offers an alternative: exporting to a NerveCenter flat file database, which is made up of two text files (*.ncdb and *.node). The flat file database files can be used on UNIX or can be imported into a database on Windows.

NOTE

The NerveCenter Server checks its connection to its database server on a regular basis. If the connection has been broken, the NerveCenter Server attempts to reestablish it. Upon reestablishing the connection, the NerveCenter Server writes its current data to the database. The retry interval is two minutes.

If, when the NerveCenter Server exits gracefully, its connection to the database server is down, it saves its current database information to an **.ASC** file in the NerveCenter DB directory and logs a message in the event log. The format of the file is **MONTHDAYYEAR_HOURMINUTE.ASC**. You can then use SerializeDB to restore this data.

You can use any number of methods to set up regularly scheduled backups, depending on your environment. Select the method that fits in best with your existing backup strategy. Here are some suggestions:

- If you use a backup tool for scheduled backups already, use it to back up the NerveCenter database, too.
- If you use a scheduling tool, create a process to back up the NerveCenter database.
- If you have a Microsoft SQL Server database, use the SQL Server backup utilities to back up the database.
- On UNIX, set up a cron job to run SerializeDB (the backup utility that ships with NerveCenter) from the command line at a regular interval.

If you need to do a manual backup in addition to your scheduled backups, see *Backing up the Database on page 209*.

NOTE
We recommend you use SerializeDB (application or command line) instead of manually copying the database file to another directory to make sure you back up the correct files and to minimize the size of the backup file.

Serialize DB exports the data in your database to a text file. When you use the SerializeDB GUI on Windows, you can choose the text file format—you can have SerializeDB export the data to a NerveCenter flat file database or a serialized ASCII file. You might want to export data from a database on Windows to a flat file database if you want to use the resulting file on a UNIX machine without having to import the data.

CAUTION
Before you back up the database, you must shut down the NerveCenter Server. Not shutting down the NerveCenter Server, could result in lost data.

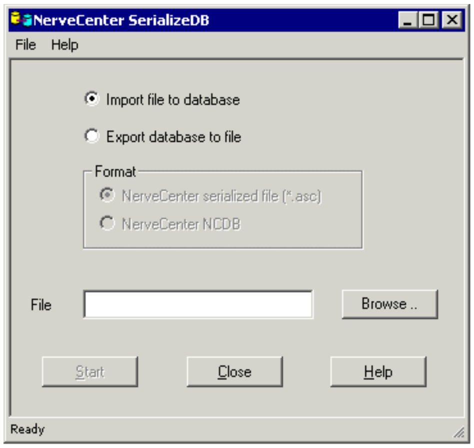
If you are backing up a UNIX flat file database, all of the files that make up the database must be in the db directory. See *Database Formats on page 204* for more information.

TO BACK UP THE DATABASE USING THE SERIALIZEDB APPLICATION

1. Do one of the following to start SerializeDB:

   - On Windows, start **SerializeDB** from the NerveCenter group in the **Start** menu.
   - In a UNIX shell, change directories to the NerveCenter bin directory and type **./serializedb &**. Then press **Enter**.

   The SerializeDB window is displayed.



2. Select the **Export database to file** radio button.
3. Specify the format by selecting one of the following radio buttons:

   - NerveCenter **serialized file (*.asc)**
   - NerveCenter **NCDB** (Windows only)

   NOTE
   If you will not use the resulting files on UNIX , we recommend you select the NerveCenter serialized file option. The ASC file is smaller, and because it is a single file, it is easier to manage.

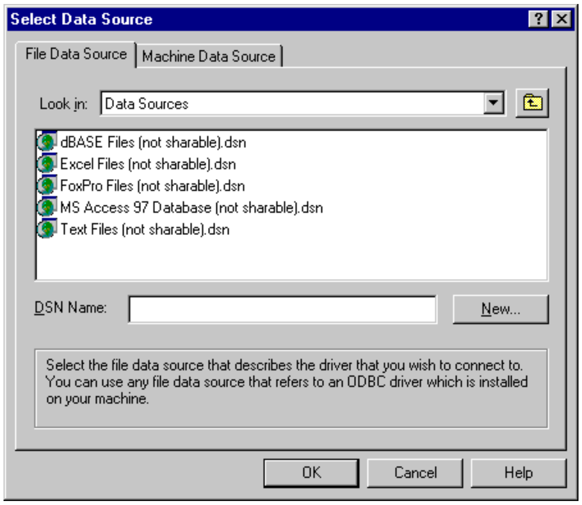
4. Do one of the following:

   - To export the data to a new file, type a name for the file (for example, **backup061503**) in the **File** field. If you do not specify a path as part of the name, SerializeDB creates the file in the NerveCenter db directory.
   - To overwrite existing data, click **Browse**, select the existing file, and click **Open**.

5. Select **Start**.

   On UNIX, SerializeDB starts exporting the data.

   On Windows, you must specify the data source for the database. The Select Data Source window is displayed.



dBASE Files (not sharable).dsn
Excel Files (not sharable).dsn
FoxPro Files (not sharable).dsn
MS Access 97 Database (not sharable).dsn
Text Files (not sharable).dsn

Select the file data source that describes the driver that you wish to connect to. You can use any file data source that refers to an ODBC driver which is installed on your machine.

   a. Select the **Machine Data Source** tab.

   b. Select the data source for your NerveCenter database. Then select **OK**.

When the data has been exported, SerializeDB displays a message letting you know that the data transfer was successful. See your Microsoft ODBC documentation for more information about selecting data sources.

When running SerializeDB from the command line, you can only export to a serialized (**.asc**) file.

Restoring the Database

If you have established a regular backup schedule and make sure you back up the database when you are doing anything that might affect the database (such as an upgrade), you'll be able to restore your data with minimal loss if you have a database problem. Here are some typical examples of situations when you'll want to restore the database:

- The machine on which the database is stored crashes. You install NerveCenter on another machine and restore your database.
- You import new behavior model objects, but discover they are causing some problems. You roll back the database so you can eliminate the problems while you troubleshoot the behavior models.
- To preserve the data in your Microsoft Access database during an upgrade, you back it up. After the upgrade, you restore your data and continue running NerveCenter.

For more information, see:

- *Restoring the Database Using the SerializeDB Application on page 213*
- *Restoring the Database from the Command Line on page 216*

NOTE

We recommend that you use SerializeDB (application or command line) instead of manually copying a backup to the db directory to make sure you import the data correctly and reestablish the connection to the database.

Restoring the Database Using the SerializeDB Application

SerializeDB imports the data in your database from a text file. When you use the SerializeDB GUI on Windows, you can have SerializeDB import the data from a NerveCenter flat file database or a serialized ASCII file.

CAUTION

Before you restore the database, you must shut down the NerveCenter Server. If you don't shut down the NerveCenter Server, you could lose data.

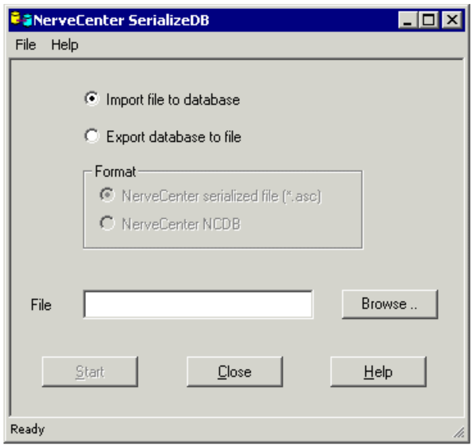
If you are restoring a UNIX flat file database, all files that make up the database must be in the db directory. See *Database Formats on page 204* for more information.

To restore the database using the SerializeDB application

1. Do one of the following to start SerializeDB:

   - On Windows, start **SerializeDB** from the NerveCenter program group in the **Start** menu.
   - In a UNIX shell, change directories to the NerveCenter bin directory and type **./serializedb &**. Then press **Enter**.

   The SerializeDB window is displayed.



2. Select the **Import file to database** radio button.

   The Format radio buttons are not available because they do not apply to importing data.

3. In the File field, specify the ***.asc** (UNIX and Windows) or ***.ncdb** file (Windows) from which you want to import data.

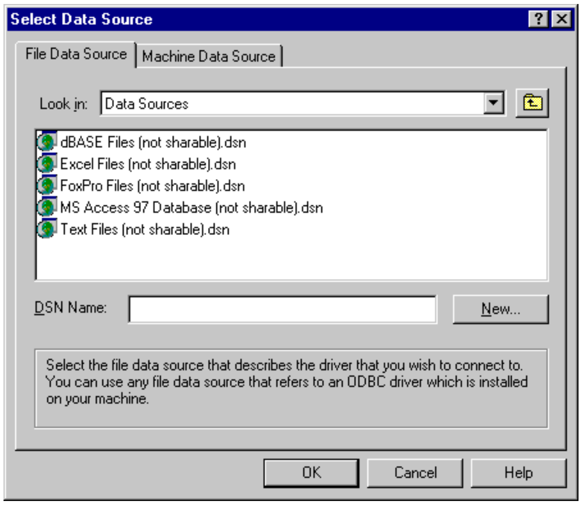
   You can also use the **Browse** button to select a file. In the Select File dialog box, select the file; then select **Open**.

   NOTE
   If you select an *.ncdb file, the file must not have more than one period (.) in the name—NerveCenter.old.ncdb, for example, would not work.

4. Select **Start**.

   On UNIX, SerializeDB starts importing the data.

   On Windows, you must specify the data source for the database. The Select Data Source window is displayed.



dBASE Files (not sharable).dsn
Excel Files (not sharable).dsn
FoxPro Files (not sharable).dsn
MS Access 97 Database (not sharable).dsn
Text Files (not sharable).dsn

Select the file data source that describes the driver that you wish to connect to. You can use any file data source that refers to an ODBC driver which is installed on your machine.

   a. Select the **Machine Data Source** tab.

   b. Select the data source for your NerveCenter database. Then select **OK**.

When the data has been imported, SerializeDB displays a message letting you know that the data transfer was successful.

Restoring the Database from the Command Line

When you run SerializeDB from the command line, SerializeDB imports the data into your database from a serialized ASCII file. You can issue SerializeDB commands from a script. For example, on UNIX, you can write a script called restorencdb that shuts down the server, imports the data from backup.asc, and restarts the server.

CAUTION

Before you restore the database, you must shut down the NerveCenter Server. If you don't shut down the NerveCenter Server, you could lose data.

If you are restoring a UNIX flat file database, all of the files that make up the database must be in the db directory. See *Database Formats on page 204* for more information.

TO RESTORE THE DATABASE USING THE COMMAND LINE

1. Enter one of the following:

   - At a DOS prompt:

     ```
     serializedb -F backupfile.asc -O FileToDB -R
     ```

     or

     ```
     serializedb -F backupfile.asc -O FileToDB -C
     "DSN=datasourcename;UID=userID;PWD=password"
     ```

     Replace *backupfile*.**asc** with the name of the file you want SerializeDB to import. If your current working directory is not the directory in which *backupfile*.**asc** is located, specify the path with the name of the file.

     - Use **-R** to have SerializeDB get the data source name, user ID, and password from the registry. You can use **-R** with a Microsoft SQL Server or Access database.
     - Use **-C "DSN=***datasourcename***;UID=***userID***;PWD=***password***"** instead of **-R** only if you are restoring a Microsoft SQL Server database. Replace *datasourcename* with the existing data source name, *userID* with the appropriate SQL Server user ID, and *password* with the corresponding password.

- In a UNIX shell:

CAUTION
On UNIX, SerializeDB requires an Xdisplay even in command line mode. Therefore, do not run SerializeDB remotely or on a headless server via TTY.

```
serializedb -F backupfile.asc -O FileToDB -R -D NerveCenter.ncdb
```

Replace *backupfile***.asc** with the name of the file you want SerializeDB to import. If your current working directory is not the directory in which *backupfile***.asc** is located, specify the path with the name of the file.

NOTE
The ***.ncdb** file must not have more than one period (.) in the name—NerveCenter.old.ncdb, for example, would not work.

2. Check to make sure the data was imported.

TIP
Use the timestamp and file size as an indicator of success. For example, if the timestamp is old or the file size is 0 kb, you need to investigate further.

If the SerializeDB does not import the data successfully, check the Event Log on Windows or the system log on UNIX for error messages.

Transferring Data between Databases

You can transfer data between databases on different platforms, between databases that have different formats, or both.

CAUTION

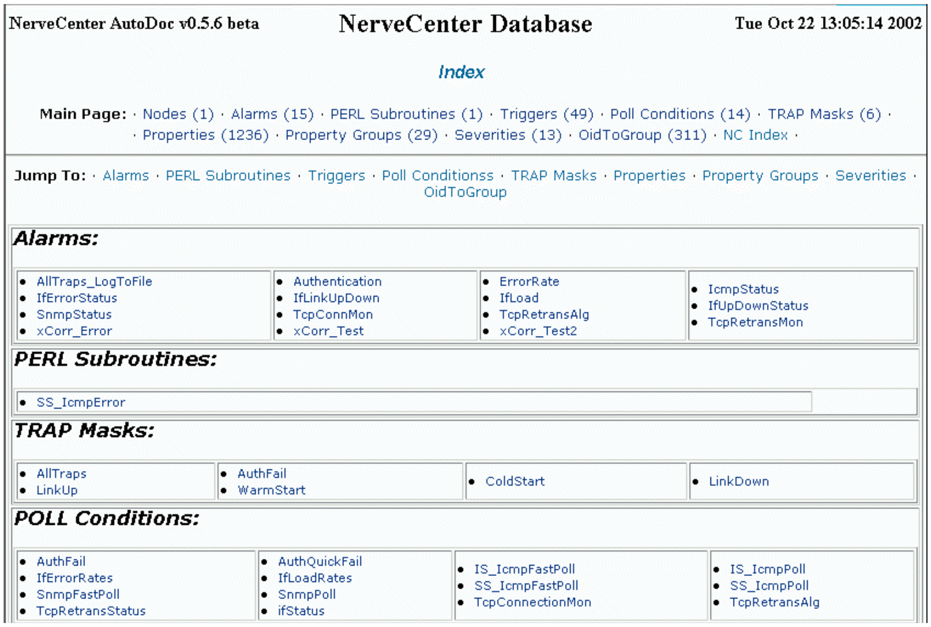
Remember, before doing anything that affects your database, including transferring data between databases, shut down the NerveCenter Server and back up the database.

To transfer data, you'll need to know how to:

- Export, or back up, data (*Backing up the Database on page 209*)
- Import, or restore, data (*Restoring the Database on page 213*)

ncdb2html.pl

NerveCenter includes a utility that converts *.ncdb and *.node files to HTML, so that you can easily know what NerveCenter objects are in your database.



NerveCenter AutoDoc v0.5.6 beta **NerveCenter Database** Tue Oct 22 13:05:14 2002

Index

Main Page: · Nodes (1) · Alarms (15) · PERL Subroutines (1) · Triggers (49) · Poll Conditions (14) · TRAP Masks (6) · · Properties (1236) · Property Groups (29) · Severities (13) · OidToGroup (311) · NC Index ·

Jump To: · Alarms · PERL Subroutines · Triggers · Poll Conditionss · TRAP Masks · Properties · Property Groups · Severities · OidToGroup

Alarms:

- AllTraps_LogToFile
- IfErrorStatus
- SnmpStatus
- xCorr_Error
- Authentication
- IfLinkUpDown
- TcpConnMon
- xCorr_Test
- ErrorRate
- IfLoad
- TcpRetransAlg
- xCorr_Test2
- IcmpStatus
- IfUpDownStatus
- TcpRetransMon

PERL Subroutines:

- SS_IcmpError

TRAP Masks:

- AllTraps
- LinkUp
- AuthFail
- WarmStart
- ColdStart
- LinkDown

POLL Conditions:

- AuthFail
- IfErrorRates
- SnmpFastPoll
- TcpRetransStatus
- AuthQuickFail
- IfLoadRates
- SnmpPoll
- ifStatus
- IS_IcmpFastPoll
- SS_IcmpFastPoll
- TcpConnectionMon
- IS_IcmpPoll
- SS_IcmpPoll
- TcpRetransAlg

FIGURE 14-3. Sample ncdb2html.pl Output

Table 14-1 details the contents of the HTML output:

TABLE 14-1. Output of ncdb2html.pl

File	Contents
ncindex.html	Summarizes the contents of your database. Provides links to all other pages. Contains the following categories: ◆ Alarms ◆ Perl subroutines ◆ Trap masks ◆ Poll conditions ◆ Triggers ◆ Properties ◆ Property Groups ◆ Severities ◆ Oid to Group

nodes.html	Details each alarm in your database, including the following information:
	◆ Node Name
	◆ Node Properties
	  ◆ Status
	  ◆ Property Group
	  ◆ Suppressible
	  ◆ Auto Delete
	◆ SNMP Properties
	  ◆ Read-Only Community
	  ◆ Read-Write Community
	  ◆ Port
	  ◆ SNMP Version
	◆ Address List

TABLE 14-1. Output of ncdb2html.pl (Continued)

File	Contents
alarms.html	Details each alarm in your database, including the following information: ◆ Status ◆ Scope ◆ Property ◆ Notes ◆ State and transition summary
perlsub.html	Lists all Perl subroutines in your database.
triggers.html	Lists all triggers in your database, including the following information: ◆ Trigger name ◆ Trigger type ◆ Where trigger is fired from
polls.html	Details each poll in your database, including the following information: ◆ Status ◆ Property ◆ Base Object ◆ Frequency ◆ Suppressible ◆ Notes ◆ Poll Condition
masks.html	Details each trap mask in your database, including the following information: ◆ Status ◆ Generic trap value ◆ Vendor specific trap number ◆ Enterprise filter (From/From Only) ◆ Simple trigger ◆ Notes ◆ Mask function
property.html	Lists all properties defined in your database and the property groups they belong to.

TABLE 14-1. Output of ncdb2html.pl (Continued)

File	Contents
propertygrp.html	Lists all property groups in your database and all properties included in each group
severities.html	Details each severity defined in your database, including the following information: ◆ Severity Name ◆ Color ◆ group ◆ level ◆ platform_level ◆ platform_name
oidtogroup.html	Lists all OIDs in your database and the MIB module the OID belongs to.

The *.ncdb and*.node files are the database format on UNIX systems. To convert Access or SQL NerveCenter databases, use the SerializeDB utility. For more information about SerializeDB, see *NerveCenter Database Utilities on page 206* or the SerializeDB help.

TO CONVERT *.NCDB AND *.NODE TO HTML

NOTE
This script requires Perl 5.6.0 or greater to execute.

1. From a command line or UNIX prompt, navigate to ***installation*/Samples/ncdb2html**.
2. Type the following command:

```
ncdb2html.pl -ncdb=path/filename.ncdb
-node=path/filename.node -h    -dir=output_directory
```

The options for this script are detailed in *Table 14-2*:

TABLE 14-2. ncdb2html.pl Options

Option	Description
-ncdb=*ncdb_file*	Required. Specify the location of ncdb file (**./NerveCenter.ncdb** is default)
-node=*node_file*	Optional. Specify the location of the node file (no default)
-dir=*output_directory*	Specify the directory to place the HTML files (current directory is the default)
-title=*title*	Specify a title for the top of each page (no default)
-h	Hide the community strings on the node page
-sh	Suppress the Open header at the top of each page
-usage	Display usage information and exit

Troubleshooting: Managing the NerveCenter Database

The following list contains some common database problems that users encounter.

The server shut down gracefully, but event log said that the database was not found. Where did the updates to the database go?

Problem: The database was not available (if, for example, the network connection between the NerveCenter server and the SQL database went down) so NerveCenter couldn't write the changes to the database.

Solution: NerveCenter serializes the database information if the database is not available. The file is date_time.asc in the db directory. While the server is still shut down, import the contents of the file when the database is available again. See *Restoring the Database on page 213*.

Running SerializeDB from the command line keeps failing

Problem: The paths to your databases or the serialized files are not specified correctly.

Solution: If you are not in the db directory, make sure you include the paths in the command line options where appropriate. See *Backing up the Database on page 209* or *Restoring the Database from the Command Line on page 216*.

SerializeDB won't work from the application or the command line

Problem: You must be an Administrator or root to run SerializeDB.

Solution: Make sure you are logged in correctly. See *Managing the Database on page 203*.

The data in the database is not what was imported from the serialized file

Problem: The server was running when you imported the file. The server overwrote the data you imported when it saved its changes to the database.

Solution: When you import or export data, make sure the server is not running. If it was, you must shut it down and reimport or export the data. See *Backing up the Database on page 209* or *Restoring the Database on page 213*.

15 Managing Management Information Bases (MIBs)

By modifying the LogMatrix NerveCenter MIB, you control which devices NerveCenter can effectively manage. The major SNMP MIB definitions and LogMatrix management application MIB definitions are already compiled in the default NerveCenter MIB. You can add other standard MIB definitions and vendors’ MIB definitions—many of which are shipped with NerveCenter—as you need them.

This chapter includes the following sections:

Section	Description
Sources of MIB Definitions on page 226	Describes the definitions that are pre-compiled in the NerveCenter MIB, the vendor definitions that are shipped with NerveCenter, and guidelines for using definitions from other sources.
Adding or Removing MIB Definitions on page 228	Describes how to add and remove references to MIB definitions from a text file that determines which MIB definitions are compiled into the NerveCenter MIB.
Compiling the NerveCenter MIB on page 231	Describes how to use the MIB compiler to compile the NerveCenter MIB.
Troubleshooting: Managing MIBs on page 233	Lists common problems users face when using MIBs.

Sources of MIB Definitions

Several MIB definitions—the LogMatrix management application definitions and the major Internet standard SNMP definitions—are already compiled into nervectr.mib, the MIB that is installed with NerveCenter.

MIB definitions from other vendors (for Cisco router support, for example) are shipped with NerveCenter. These definitions are not pre-compiled into the NerveCenter MIB because incorporating all the definitions would increase the size of the MIB and could affect performance unnecessarily. You may add the definitions that are appropriate for your environment.

You can also add MIB definitions you've gotten from other sources.

For more information, see:

- *Definitions Compiled Into the NerveCenter MIB on page 226*
- *Definitions Not Compiled Into the NerveCenter MIB on page 227*
- *Definitions From Other Sources on page 228*

Definitions Compiled Into the NerveCenter MIB

The definitions that are already compiled into the NerveCenter MIB are the LogMatrix management application definitions and the major Internet standard SNMP definitions. The definitions are in subdirectories in the mib directory (mibs on UNIX).

Table 15-1 lists the subdirectories that contain these definitions and includes descriptions of each.

TABLE 15-1. Definitions Compiled into the MIB

This directory...	Includes definitions for...
nsmg	NerveCenter management applications
standard-v1	Internet standard RFC SNMPv1 MIB definitions (converted to SNMPv2 compliance)
standard-v2	Internet standard RFC SNMPv2 MIB definitions
standard-v3	Internet standard RFC SNMPv3 MIB definitions

Definitions Not Compiled Into the NerveCenter MIB

Based on your environment, you decide which vendor definitions to compile into the NerveCenter MIB. The definitions are in subdirectories in the mib\vendors directory (mibs/vendors on UNIX).

The following table lists the vendor subdirectories that contain these definitions and includes descriptions to help you decide which ones you need.

TABLE 15-2. Definitions Not Compiled into MIB

This directory...	Includes definitions for...
3com	ECS Ethernet hubs
att	Smarthub product line
baynetworks	System 3000/5000 concentrators, Linkswitch Ethernet switches, and ATM Ethercell switches
chipcom	Networking product line
cisco	Router support of IOS 9.x, 10.x, and 11.x
compaq	Server, storage, and UPS systems
dec	DIGITAL ELAN networking devices
epix	ExpressLink product line
fibermux	Crossbow product line
microsoft	Servers and Internet Servers
nat	Network probes
ods	Ethernet and FDDI devices
retix	Remote bridge and router product lines
sun	UNIX SNMP agents
ungermanbass	Access/One and Ethernet concentrator products
wellfleet	Router support of 6.x/7.x SNMP agents
xyplex	Terminal servers

Definitions From Other Sources

You might want to add other MIB definitions to manage devices that are not supported by the MIBs shipped with NerveCenter. You can add MIB definitions from other sources as long as each one:

- Is a complete ASN.1 module definition that conforms to the standards specifications in RFC1902, *Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)*, RFC1903, *Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)*, and RFC1904, *Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)*. You can find these RFCs and others on the Web.

CAUTION

Before NerveCenter can compile a mib it must conform to the ASN.1 module definition standard. See the section *Troubleshooting ASN.1 files on page 271*.

- Does not have the same name as a MIB definition already compiled into your NerveCenter MIB.

Also, remember that your devices must run SNMP agents from your vendor that support the new MIB definitions.

We recommend that you store additional definitions in the mib\user directory (you can create a user directory in the mibs directory on UNIX). Then, when you install a new version of NerveCenter, you can make a copy of this directory to preserve your additional third-party MIBs.

Adding or Removing MIB Definitions

The MIB compiler (mibcomp) compiles the MIB definitions referenced in a text file. You can add references to or remove them from this file to control what definitions are compiled into the NerveCenter MIB.

Add MIB definitions if you want to monitor additional devices. Remove MIB definitions if you are no longer monitoring the devices supported by those definitions. (Removing the definitions is not required; however, the NerveCenter MIB will be smaller, which can improve client performance and will make managing the MIB easier.)

If you are using NerveCenter with a network management platform and both applications are monitoring the same agents, make sure both applications are using the correct MIBs for those agents.

See *Sources of MIB Definitions on page 226* for a list of MIB definitions that are shipped with NerveCenter and what standards MIB definitions are not shipped with NerveCenter must follow.

NOTE

NerveCenter installs the file nervectr50.mib and mibcomp50.txt. These are copies of the default nervectr.mib and mibcomp.txt in case you need to revert to an unchanged MIB definitions.

The MIB compiler (mibcomp) compiles the MIB definitions referenced in a text file. You can add references to this file to control what definitions are compiled into the NerveCenter MIB.

CAUTION

If a MIB module depends on information in another MIB module you must reference it before the module that requires the information in mibcomp.txt. That is, if mymibA depends on mymibB, you must include mymibB before mymibA in mibcomp.txt.

If dependant MIB module references are out of order, you receive an error similar to the following:

```
8: standard-v2/rfc1573b.asn1 mgrtool.exe: process_data(), Couldn't
find parent: interfaces
mibcomp: unable to compile and resolve standard-v2/rfc1573b.asn1
```

In this example, **rfc1573b.asn1** depends upon another mib to define *interfaces*. You must find the mib that defines interfaces and include it before **rfc1573b.asn1** in **mibcomp.txt**. To find a MIB that defines interfaces, you search the other MIB modules looking for the following:

```
interfacesOBJECT IDENTIFIER ::= { mib-2 2 }
```

Now you can name the file that contains this line, in this case **rfc1213.asn1**, before **rfc1573b.asn1**.

CAUTION

All SMIv1 compliant MIB modules must be included in **mibcomp.txt** before the line

```
#* * * End of SMIv1 / SNMPv1 MIB Modules * * *
```

All SMIv2 compliant MIB modules must be referenced after this line.

You can tell the difference between a SMIv1 module and an SMIv2 module by searching the *.asn1 file for “MAX-ACCESS.

Sample SMIv1 SysUpTime definition	Sample SMIv2 SysUpTime definition
sysUpTime OBJECT-TYPE	sysUpTime OBJECT-TYPE
SYNTAXTimeTicks	SYNTAXTimeTicks
ACCESS read-only	MAX-ACCESS read-only
STATUS mandatory	STATUS current
DESCRIPTION	DESCRIPTION
“The time (in hundredths of a second) since the network management portion of the system was last re-initialized.”	“The time (in hundredths of a second) since the network management portion of the system was last re-initialized.”
::= { system 3 }	::= { system 3 }

Any use of ACCESS within a *.asn1 named in the SMIv2 area of mibcomp.txt causes an error. Any use of MAX-ACCESS within a *.asn1 named in the SMIv1 area of mibcomp.txt causes an error.

TO MODIFY THE MIBCOMP TEXT FILE

1. If you previously modified **mibcomp.txt** or **nervectr.mib**, make back-up copies of them in case you want to recover your changes.

   NOTE
   You can modify these files directly; they will not be overwritten during the next installation.

2. In a text editor, open the file.
3. To add definitions, do one of the following:
   - Remove the pound sign (#) in front of definitions that are currently commented out.
   - Add lines for new definitions. Include the directory and the name of the file. For example:

   ```
   users/companyXYZ.asn1
   ```

   NOTE
   On Windows, use backslashes. On UNIX, you must use slashes.

4. To remove definitions, do one of the following:

   - Comment out definitions by adding a pound sign (#) in front of the appropriate lines. For example:

```
# xyplex
#
# vendors/xyplex/xyplex.asn1
```

   - Delete the appropriate lines.

5. Save and close the file.

You have updated the file that includes references to the MIB definitions you need. Now, you must compile the NerveCenter MIB using the file you just created. See *Compiling the NerveCenter MIB on page 231*.

Compiling the NerveCenter MIB

After you modify the mibcomp.txt file, you must compile the NerveCenter MIB. The MIB compiler updates the MIB to support the types of devices you want to monitor.

- **mibcomp** is NerveCenter's MIB compiler. (see *mibcomp on page 296* for syntax and options)
- **mibTool** is a new Java-based MIB compiler introduced with NerveCenter 5.1. (see *mibTool on page 298* for syntax and options)

To compile the MIB

1. At a command prompt, change to the mib directory (mibs on UNIX).
2. Type one of the following commands and press **Enter**.

```
..\bin\mibcomp mibcomp.txt
..\bin\mibTool -mibcomp mibcomp.txt
```

As the compiler runs, it displays a series of messages. If a problem needs your attention, the compilation fails. You must resolve the error and re-compile. The error is displayed on the last line on your screen. You will have to do this more than once if you have several errors. Errors are generally syntax problems in third-party MIBs or MIBs that do not comply with the RFC specification. For tips on solving compilation errors, see *Troubleshooting: Managing MIBs on page 233*.

3. If you did not compile the MIB on the machine that is running the NerveCenter Server, copy it to that machine.

CAUTION

The MIB must be located locally on the machine hosting NerveCenter Server. Configure security on the server machine appropriately so no unauthorized users can overwrite or change the MIB. On UNIX, set permissions to read-only using the following command: `chmod 444 nervectr.mib`. On Windows, make the .mib file read-only or set its directory permission to read-only.

You have updated the MIB that includes the definitions you need. Now, you must restart the NerveCenter server to load this MIB. If your network management platform uses the NerveCenter MIB, you'll need to update that information in your network management platform. See the documentation for your network management platform for details.

If you added new definitions, you may want to create new property groups that contain properties for the new MIB base objects. See *Defining Property Groups and Properties in Designing and Managing Behavior Models*.

Troubleshooting: Managing MIBs

Following are some common problems that users encounter when updating NerveCenter MIBs.

As mentioned in *Adding or Removing MIB Definitions on page 228*, MIB modules are dependant on information in other MIB modules. Although a MIB module may compile by itself, there may be problems with dependencies with other MIB modules. This section gives suggestions on how to solve compilation errors.

TIP

If MibComp finds a dependency or definition error, it will repeat the error for each MIB module encountered. To find where the problem begins, run MibComp with the **-clean** and **-trace** arguments and note when the problem first occurs.

A MIB won't compile

Problem: The vendor ASN1 definitions that you added are not formatted correctly.

Solution: Check for spelling and format.

See *Troubleshooting ASN.1 files on page 271*.

Problem: The mibcomp.txt file is not formatted correctly.

Solution: Check to make sure the appropriate lines are included in the file and not commented out.

See *Adding or Removing MIB Definitions on page 228*.

Problem: The ASN1 file does not exist in the directory referenced in the mibcomp.txt file.

Solution: Move the file to the specified directory or correct the path in the mibcomp.txt file.

Problem: You must run the command from the mib directory (mibs on UNIX).

Solution: Change to the correct directory and run the mibcomp command again.

See *Compiling the NerveCenter MIB on page 231*.

Problem: User or file permissions are not set correctly, so mibcomp cannot create or update the MIB file.

Solution: Check user or file permissions and correct them, if necessary. See your operating system documentation.

Problem: Redefinitions of MIB entity enumerations across SNMPv1, v2c, and v3.

Example: WARNING: enumeration conflicts during merge for ifType (continuing)

Solution: These warnings can be safely ignored. Some values, such as ifType or ifOperStatus, have been redefined for different SNMP versions.

Problem: More than one MIB module has defined entities with the same name.

Example: /opt/OSInc/bin/mgrtool: check_names: Duplicate name with different OIDs: ds1 OID1: 1.3.6.1.2.1.10.18, OID2: 1.3.6.1.3.2 continuing (since -i option was used)

Solution: First, find which MIB modules are defining the duplicated name. When MibComp finds a second definition, it creates an error message similar to the above example. Next, you must research the MIB modules to see if either module has been replaced with a compliant MIB module.

Problem: A MIB module depends upon a definition form another MIB module.

Example: find_type(): unknown type: SnmpAdminString
mibcomp: unable to compile and resolve standard-v3/v3-tgt.my

Solution: Look at the IMPORTS clause of the module that produced the error. In the example, **v3-tgt.my** declares that SnmpAdminString is to be imported from SNMP-FRAMEWORK-MIB. A search of the modules shows that SNMP-FRAMEWORK-MIB is defined in **v3-arch.my**. Therefore, you must name **v3-arch.my** before **v3-tgt.my**.

Problem: More than one MIB module provides a name for an OID.

Example: /opt/OSInc/bin/mgrtool: Warning: Duplicate OIDs with different names.
The first name will appear before the second name.
OID: 1.3.1.4.1.9.7.99999.2, First name: cwRsrcPartCapabilityRpmV2R0160,
Second name: ciscoWanModuleCapabilityV2R00

Solution: Often this is not a problem but should be investigated. In the example, two different Cisco MIB modules state two different names for the same OID. Which device and version of IOS is encountered at run-time will likely resolve which way the OID will be handled. Check which devices you are currently running and whether you need to have the MIB modules with the duplicate names **mibcomp.txt**. A review of your Cisco devices may show that the MIB module that names the first occurrence is not needed.

A MIB won’t load

Problem: You must be a member of ncadmins on UNIX or NerveCenter Admins on Windows to load a MIB.

Solution: Make sure you have appropriate group membership.

Problem: The MIB must be located on the server machine.

Solution: Copy the MIB to the server machine and try loading it again.

See *Compiling the NerveCenter MIB on page 231*.

Problem: The name or location of the MIB has changed.

Solution: Update the name and location of the MIB and try loading it again.

Error Messages

A

This appendix explains the error and information messages that you might encounter while using NerveCenter. Possible causes and solutions for the errors are included.

This appendix includes the following sections:

Section	Description
User Interface Messages on page 236	Explains where error messages appear as well as the different types of error messages.
Error Messages on page 238	Lists the error messages and possible solutions.

User Interface Messages

All NerveCenter error messages are written to the Event Log. To view messages in the Event Log:

- Windows: Run the Event Viewer and display the Application log. Each error message is listed as a line in the log.
- UNIX: Read the file /var/adm/messages with a text editor or a command such as more.

Each error description is formatted in the following way:

```
Category error_message_number: message: [code_number]
```

Each message is assigned a category, which has a corresponding number. The line listed in the log uses a number to indicate a category, as follows:

TABLE A-1. Error Message Categories

Number	Category
1	NC Server Manager
2	NC Alarm Manager
3	NC Trap Manager
4	NC Poll Manager
5	NC Action Manager
6	NC Protocol Manager
7	NC PA Resync Manager
8	NC Service
9	NC Inform NerveCenter Manager
10	NC OpC Manager
11	NC LogToFile Manager
12	NC FlatFile Manager
13	NC Alarm Filter Manager
14	NC Deserialize Manager
15	NC LogtoDB Manager
16	NC DB Manager
17	NC Inform OV

The error message number indicates the error type. The error numbers are organized as follows:

TABLE A-2. Error Message Numbers

Number Range	Type of Error
0-999	Users should call customer support.
1000-1999	User can resolve the problem.
2000-2999	User is warned of an event.
3000-3999	User is given an informational message.

The error messages are explained in the following sections:

- *Action Manager Error Messages on page 239*
- *Alarm Filter Manager Error Messages on page 243*
- *Deserialize Manager Error Messages on page 243*
- *Flatfile Error Messages on page 243*
- *Inform NerveCenter Error Messages on page 244*
- *Inform OV Error Messages on page 244*
- *LogToDatabase Manager Error Messages on page 246*
- *LogToFile Manager Error Messages on page 247*
- *Poll Manager Error Messages on page 247*
- *Protocol Manager Error Messages on page 247*
- *PA Resync Manager Error Messages on page 248*
- *Server Manager Error Messages on page 250*
- *Trap Manager Error Messages on page 254*
- *NerveCenter installation Error Messages (UNIX) on page 255*
- *OpenView Configuration Error Messages (UNIX) on page 257*

A Error Messages

Error Messages

The following tables list particular error messages that may occur when operating NerveCenter. For an explanation of what types of error messages exist and where error messages appear, see the section *User Interface Messages on page 236*.

The messages include:

- *Action Manager Error Messages on page 239*
- *Alarm Filter Manager Error Messages on page 243*
- *Deserialize Manager Error Messages on page 243*
- *Flatfile Error Messages on page 243*
- *Inform NerveCenter Error Messages on page 244*
- *Inform OV Error Messages on page 244*
- *LogToDatabase Manager Error Messages on page 246*
- *LogToFile Manager Error Messages on page 247*
- *Poll Manager Error Messages on page 247*
- *Protocol Manager Error Messages on page 247*
- *PA Resync Manager Error Messages on page 248*
- *Server Manager Error Messages on page 250*
- *Trap Manager Error Messages on page 254*
- *NerveCenter installation Error Messages (UNIX) on page 255*
- *OpenView Configuration Error Messages (UNIX) on page 257*

Action Manager Error Messages

Following is a list of Action Manager error messages.

TABLE A-3. Action Manager Error Messages

#	Error	Resolution
1	Action Manager Initialization failed with send trap socket	N/A
3	Send trap action: CreateTrapRequest failed	N/A
4	Send trap action: Send trap failed	N/A
500	Socket Error: *value*	N/A
501	<system call> failed while launching Application handler : <error message>	N/A
1001	Action Manager connect to database failed	Check NerveCenter database. Check ODBC connection string.
1002	InitializePlatformSocket failed for *value*	Use the Administrator to check the configuration settings for NetNodeNotify.
1004	Can't open database	Check NerveCenter database. Check ODBC connection string.
1005	No connection string for Log to Database action	Check ODBC connection string.
1006	Reconfiguration: InitializePlatformSocket failed for *value*	Check Notify page in NC Admin.
1010	Log to Event View error: RegisterEventSource for *value* failed with error code *value*	Check system configuration.
1011	Log to Event View error: ReportEvent failed with error code *value*	Check system configuration.
1012	Socket Creation Failed in InitSmtpSocket With Error = *value*	Check socket resource on the computer.
1013	Protocol Bind Failed in InitSmtpSocket With Error = *value*	Check TCP/IP configuration.
1014	Connect to SMTP Host Failed in InitSmtpSocket With Error=*value*	Use the Administrator to check the configuration settings for SMTP host name.
1015	Ioctlsocket Failed (Setting Non-Blocking Mode) in InitSmtpSocket With Error=*value*	Check TCP/IP configuration.
1016	Send Packet Failed in SendSmtpPacket With Error= *value*	Check SMTP server.

TABLE A-3. Action Manager Error Messages (Continued)

#	Error	Resolution
1017	Receive Packet Failed in RecvSmtpPacket for %1 With Error= *value*	Check SMTP server.
1018	Received Unexpected Response= *value* in RecvSmtpPacket	Check SMTP server.
1019	Log to Database error: Database connection not open	Check NerveCenter database. Check SQL Server.
1020	Log to Database error: can not open log table	Check NC_Log table in NerveCenter database.
1021	Log to Database exception: *value*	Check NerveCenter database. Check SQL Server. Check NC_Log table in NerveCenter database.
1022	Logging to a File error: No filename presented to Log To File action.	Make sure there is a file name associated with LogToFile action for alarm transitions.
1023	Logging to a File error: Unable to Write LogFile: *value* Error Code = *value*.	Check security on file system. Make sure the file is writable.
1024	Logging to a File error: Unable to Create LogFile: *value* Error Code = *value*.	Check security on file system. Make sure the file is writable.
1025	Logging to a File error: Unable to Seek EOF for LogFile: *value* Error Code = *value*	Check security on file system. Make sure the file is writable.
1026	Logging to a File error: Unable to Truncate LogFile.	Delete the file or repair the file format.
1027	Could Not Logoff from MAPI *value*, Error=*value*	Check MAPI service in the system.
1028	Could Not Load MAPI32.DLL.	Search mapi32.dll in the system and ensure sure it is in the system path.
1029	Could Not Get MAPILogon Address.	Check mapi32.dll in the system and ensure it is a good version.
1030	Could Not Get MAPILogoff Address.	Check mapi32.dll in the system and ensure it is a good version.
1031	Could Not Get MAPISendMail Address.	Check mapi32.dll in the system and ensure it is a good version.
1032	Could Not Logon to MAPI *value*, Error=*value*.	Check MAPI configuration and ensure to have created the profile.
1033	Could Not SendMail to MAPI *value*, Error=*value*.	Check MAPI configuration and ensure to have created the profile.
1034	Paging action error: Dial failed.	Check modem configuration.

TABLE A-3. Action Manager Error Messages (Continued)

#	Error	Resolution
1035	Running an NT Command error: No Command Presented to Run Command.	Make sure there is a command associated with all Windows Command actions specified for alarm transitions.
1036	Running an NT Command error: Command *value* Completed with ReturnCode *value*	Check command line.
1037	Command action *value* failed : Application handler *value* was killed	NCServer will bring it up for the next Command action
1038	Command action <action> failed : *value*	If error says “Too many open files” close some open files. If error says “fork failure” close some applications.
1039	Unable to launch Application handler: *value*	If error says “Too many open files” close some open files. If error says “fork failure” close some applications.
1040	Perl subroutine *value* failed: *message*	
1500	The connection to *value* was closed	
1505	*value*. The address is already in use	Make sure you are not running two instances of the same application on the same machine.
1506	*value*. The connection was aborted due to timeout or other failure	Make sure the physical network connections are present.
1507	*value*. The attempt to connect was refused	Make sure the server is running on the remote host.
1508	*value*. The connection was reset by the remote side	Make sure the remote peer is up and running.
1509	*value*. A destination address is required	A destination address or host name is required.
1510	*value*. The remote host cannot be reached	Make sure the routers are working properly.
1511	*value*. Too many open files	Close any open files.
1512	*value*. The network subsystem is down	Reboot the machine.
1513	*value*. The network dropped the connection	Make sure the peer is running and the network connections are working.
1514	*value*. No buffer space is available	This might be because you are running several applications, or an application is not releasing resources.

TABLE A-3. Action Manager Error Messages (Continued)

#	Error	Resolution
1515	*value*. The network cannot be reached from this host at this time	Make sure the routers are functioning properly.
1516	*value*. Attempt to connect timed out without establishing a connection	Make sure the machine is running and on the network.
1517	*value*. The host cannot be found	Make sure you can ping the host. Check your hosts file or DNS server.
1518	*value*. The network subsystem is unavailable	Make sure the network services are started on machine.
1519	*value*. Invalid host name specified for destination	The host name cannot be resolved to an IP address. Enter the name to the hosts file or DNS server.
1520	*value*. The specified address in not available	Make sure the host name is not zero—try pinging the host.
2001	Command line too long: *value*	Check the Windows Command Action. Command line exceeds maximum allowed length of 2048 characters.
2002	Send trap action failed for alarm *alarm name* due to the following reason: *string*	Check the source or destination host name. Check the enterprise. If this action was not caused by a trap, it will fail if the enterprise is $P. Check to see that the varbinds are legal for the currently loaded MIB.
2003	Tapi initialize failed, paging will not work	Check the comm port/modem configuration and check the tapi32.dll version.
2004	Empty host for SMTP mail	If SMTP actions are used, use the Administrator to enter the SMTP mail host name.
2005	Empty profile for MAPI, MS Mail will not work	If MS mail actions are used, use the Administrator to enter the SMTP mail host name.
2006	Fire Trigger Action error: Invalid node name: *value*	A node name was specified directly in an action and that node doesn't exist in the system.
2007	Fire Trigger Action error: Invalid property name: *value*	A property was specified directly in an action and that property doesn't exist in the system.
2008	Fire Trigger Action error: Invalid subobject: *value*	A subobject was specified directly in an action and that subobject doesn't exist in the system.

TABLE A-3. Action Manager Error Messages (Continued)

#	Error	Resolution
2010	Error Sending SMTP Mail. *Value* messages may have been lost.	

Alarm Filter Manager Error Messages

Following is a list of Alarm Filter Manager error messages.

TABLE A-4. Alarm Filter Manager Error Messages

#	Error	Resolution
1	Lookup failed on line number *value* in File *value*.	
3001	Alarm Filter Manager Initialization successfully finished	

Deserialize Manager Error Messages

Following is a list of Alarm Filter Manager error messages.

TABLE A-5. Deserialize Manager Error Messages

#	Error	Resolution
1	Lookup failed on line number *value* in File *value*.	
3001	Deserialize Thread Manager Initialization successfully finished	

Flatfile Error Messages

Following is a list of Flatfile Manager error messages.

TABLE A-6. Flatfile Manager Error Messages

#	Error	Resolution
1	Lookup failed on line number *value* in File *value*.	
3001	Flat File Initialization successfully finished	

Inform NerveCenter Error Messages

Following is a list of Inform NerveCenter Manager error messages.

TABLE A-7. Inform NerveCenter Manager Error Messages

#	Error	Resolution
1	Lookup failed on line number *value* in File *value*.	
3001	InformNC Manager Initialization successfully finished	

Inform OV Error Messages

Following is a list of Inform OV Manager error messages.

TABLE A-8. Inform OV Manager Error Messages

#	Error	Resolution
2	ReceiveHandShakeResponse FALSE byte not correct.	N/A
500	Socket Error: *value.*	N/A
501	<system call> failed while launching Application handler : <error message>.	N/A
1002	InitializePlatformSocket failed for *value.*	Use the Administrator to check the NetNodeNotify configuration.
1003	No platform host for InformOV.	Use the Administrator to check the NetNodeNotify configuration.
1006	Reconfiguration: InitializePlatformSocket failed for *value.*	Check Notify page in the Administrator.
1007	CInformOVEventSocket::Init() failed with invalid operation: *value.*	Use the Administrator to check the NetNodeNotify configuration.
1039	Unable to launch Application handler: *value.*	If error says “Too many open files” close some open files. If error says “fork failure” close some applications.
1040	Perl subroutine *value* failed: *message.*	
1500	The connection to *value* was closed.	
1505	*value*. The address is already in use.	Make sure you are not running two instances of the same application on the same machine.

TABLE A-8. Inform OV Manager Error Messages (Continued)

#	Error	Resolution
1506	*value*. The connection was aborted due to timeout or other failure.	Make sure the physical network connections are present.
1507	*value*. The attempt to connect was refused.	Make sure the server is running on the remote host.
1508	*value*. The connection was reset by the remote side.	Make sure the remote peer is up and running.
1509	*value*. A destination address is required.	A destination address or host name is required.
1510	*value*. The remote host cannot be reached.	Make sure the routers are working properly.
1511	*value*. Too many open files.	Close any open files.
1512	*value*. The network subsystem is down.	Reboot the machine.
1513	*value*. The network dropped the connection.	Make sure the peer is running and the network connections are working.
1514	*value*. No buffer space is available.	This might be because you are running several applications, or an application is not releasing resources.
1515	*value*. The network cannot be reached from this host at this time.	Make sure the routers are functioning properly.
1516	*value*. Attempt to connect timed out without establishing a connection.	Make sure the machine is running and on the network.
1517	*value*. The host cannot be found.	Make sure you can ping the host. Check your hosts file or DNS server.
1518	*value*. The network subsystem is unavailable.	Make sure the network services are started on machine.
1519	*value*. Invalid host name specified for destination.	The host name cannot be resolved to an IP address. Enter the name to the hosts file or DNS server.
1520	*value*. The specified address in not available.	Make sure the host name is not zero—try pinging the host.
2001	Command line too long: *value.*	Check the Windows Command Action. Command line exceeds maximum allowed length of 2048 characters.
2006	Fire Trigger Action error: Invalid node name: *value.*	A node name was specified directly in an action and that node doesn't exist in the system.
2007	Fire Trigger Action error: Invalid property name: *value.*	A property was specified directly in an action and that property doesn't exist in the system.
2008	Fire Trigger Action error: Invalid subobject: *value.*	A subobject was specified directly in an action and that subobject doesn't exist in the system.

TABLE A-8. Inform OV Manager Error Messages (Continued)

#	Error	Resolution
2009	Inform OV send Packet Failed for platform socket *value*.	
3001	Inform OV Manager Initialization successfully finished.	
3002	CInformOVEventSocket::OnClose with code *value*.	

LogToDatabase Manager Error Messages

Following is a list of Log to Database Manager error messages.

TABLE A-9. Log to Database Manager Error Messages

#	Error	Resolution
1002	Initialization failed.	Check WriteBuiltInTriggers.
1100	Unknown database exception.	Check NerveCenter database. Log segment might be full.
1101	Failed to connect to database.	Check NerveCenter database. Check ODBC connection string.
1102	Failed to connect to database.	Check NerveCenter database. Check ODBC connection string.
1103	Version table validation failed. NC_Version table doesn't exist in database.	
1104	Write to database failed.	Log segment might be full or the database might have gone down.
1203	Can't enable discovery model.	Check the alarm table and the state of alarms (off or on).
3001	Database Thread Initialization successfully finished.	
3002	The database state has changed. Either it has gone down or come up.	

LogToFile Manager Error Messages

Following is a list of Log to File Manager error messages.

TABLE A-10. Log to File Manager Error Messages

#	Error	Resolution
1	Lookup failed on line number *value* in File *value*.	
3001	LogToFile Manager Initialization successfully finished	

Poll Manager Error Messages

Following is a list of Poll Manager error messages.

TABLE A-11. Poll Manager Error Messages

#	Error
3001	Poll Manager Initialization successfully finished
3002	CPollManagerWnd:OnPollOnOff, PreCompild of PollEvent with Poll Id %ld failed

Protocol Manager Error Messages

Following is a list of Protocol Manager error messages.

TABLE A-12. Protocol Manager Error Messages

#	Error	Resolution
1	Building copy of node list failed.	N/A
2	Building copy of poll property list failed.	N/A
3	I.nitialization of protocol methods failed	N/A
4	Initialization of ping socket failed.	N/A
5	Creation of SNMP socket failed, socket error code: %d	N/A
6	Error in ping socket: %s	N/A
7	Error in ping socket: create socket failed.	N/A
8	Error in ping socket: async select failed.	N/A

TABLE A-12. Protocol Manager Error Messages (Continued)

#	Error	Resolution
1000	Looking for the %s key in the configuration settings.	Use the Administrator to enter the SNMP values in the configuration settings.
1001	Ncuser user ID is not found.	Add ncuser user ID to your system.
3000	Initialization successfully finished.	N/A
3001	Invalid value in configuration settings for SNMP retry interval, using default of 10 seconds.	Use the Administrator to enter an SNMP retry interval.
3002	Invalid value in configuration settings for number of SNMP retries, using default of 3 retries.	Use the Administrator to enter a number of SNMP retries.
3003	Invalid value in configuration settings for default SNMP port, using default of 161.	Use the Administrator to enter the default SNMP port number.

PA Resync Manager Error Messages

Following is a list of PA Resync Manager error messages.

TABLE A-13. PA Resync Manager Error Messages

#	Error	Resolution
1	Error getting local host name for encoding resync request, socket error code: %d	N/A
2	Encoding resync request failed	N/A
3	Sending resync request failed with zero bytes sent	N/A
4	Sending resync request failed: %s	N/A
5	Memory allocation error, trying to notify of connection status	N/A
6	Memory allocation error, creating node list	N/A
7	Memory allocation error, creating a resync node	N/A
8	Parent status not sent during resync	
10	Parents not computed during resync with map host. Check OVPA. OVPA database must have nc host node.	
500	Socket Error: (%d)	

TABLE A-13. PA Resync Manager Error Messages (Continued)

#	Error	Resolution
1000	Error looking for the %s key in the NerveCenter configuration settings	Use the Administrator to enter configuration settings.
1001	Attempt to connect to %s on port %d failed: %s	Make sure the platform host is up and running and that the name exists in the hosts file.
1002	Resync connection attempt failed: %d	Make sure the platform host is up and the platform adapter is running.
1500	The connection to % was closed	
1501	Send failed with zero bytes sent	
1505	%s. The address is already in use	Make sure you are not running two instances of the same application on the same machine.
1506	%s. The connection was aborted due to timeout or other failure	Make sure the physical network connections are present.
1507	%s. The attempt to connect was refused	Make sure the server is running on the remote host.
1508	%s. The connection was reset by the remote side	Make sure the remote peer is up and running.
1509	%s. A destination address is required	A destination address or host name is required.
1510	%s. The remote host cannot be reached	Make sure the routers are working properly.
1511	%s. Too many open files	Close any open files.
1512	%s. The network subsystem is down	Reboot the machine.
1513	%s. The network dropped the connection	Make sure the peer is running and the network connections are working.
1514	%s. No buffer space is available	This might be because you are running several applications, or an application is not releasing resources.
1515	%s. The network cannot be reached from this host at this time	Make sure the routers are functioning properly.
1516	%s. Attempt to connect timed out without establishing a connection	Make sure the machine is running and on the network.
1517	%s. The host cannot be found	Make sure you can ping the host, check you hosts file or DNS server.
1518	The network subsystem is unavailable	Make sure the network services are started on machine.
1519	%s. Invalid host name specified for destination	The host name cannot be resolved to an IP address. Enter the name to the hosts file or DNS server.

TABLE A-13. PA Resync Manager Error Messages (Continued)

#	Error	Resolution
1520	The specified address in not available	Make sure the host name is not zero. Try pinging the host.
3000	initialization successfully finished	N/A
3001	Node resync from map host was not requested because either host name or port number is missing	If you are trying to disable a connection to the platform adapter, then this message is OK. If you want to be connected to the platform adapter, then use the Administrator to check the map host settings.
3500	Connection to %s was successful	N/A

Server Manager Error Messages

Following is a list of Server Manager error messages.

TABLE A-14. Server Manager Error Messages

#	Error	Resolution
1	OLE initialization failed. Make sure that the OLE libraries are the correct version.	N/A
2	Perl create failed.	N/A
3	Initialization of *value* manager thread failed.	N/A
4	Failed to restore MibDirectory in configuration settings.	N/A
5	Failed to open configuration settings while trying to restore mib information.	N/A
6	Discrepancy in data. File: SERVER_CS.CPP, Line: *value.*	N/A
10	Conflict in data. File: SERVER_CS.CPP, Line: *value.*	N/A
11	Internal Error. File: SERVER_CS.CPP, Line: *value.*	N/A
20	Cannot read configuration settings value: Bind.	N/A
21	Cannot connect to Tcpip configuration settings information.	N/A
22	Cannot read configuration settings value: IPAddress.	N/A
23	Couldn't find *value* in map.	N/A

TABLE A-14. Server Manager Error Messages (Continued)

#	Error	Resolution
24	Error while reading database. Poll/Mask:*value* uses a simple trigger that doesn't exist in database.	N/A
25	Please report error number *value* to technical support.	N/A
26	User validation failed: Unable to communicate with ncsecurity process :*value*.	~
1001	Windows sockets initialization failed.	Install TCP/IP.
1002	Initialization failed, cannot find ncperl.pl.	Check NCPerl.pl location.
1003	Failed to open MIB: *value*.	Check MIB location.
1004	Failed to parse MIB.	Invalid MIB. Check configuration to see if the correct MIB is specified.
1010	Failed to validate poll: *value*. The poll will be turned off.	Check the poll condition using the Client Application.
1100	*value* (database error).	Try to resolve using the message. If not, call support.
1101	Failed to connect to database. ODBC Connection String in configuration settings is invalid or can't find database server.	Use InstallDB to re-create the ODBC connection string.
1102	Failed to connect to database. ODBC Connection String in configuration settings is empty.	Use InstallDB to re-create the ODBC connection string.
1103	Version table validation failed. NC_Version table doesn't exist in database.	Upgrade the NerveCenter database.
1200	Failed to open configuration settings while trying to restore mib information.	Use the NerveCenter Administrator to check the configuration settings. Invalid key is likely.
1201	Updated License key is invalid.	An invalid license key was entered. Check the key.
1202	Cannot connect to configuration settings.	Use the NerveCenter Administrator to check the configuration settings. Invalid key is likely.
1203	Cannot open key *value*.	Use the NerveCenter Administrator to check the configuration settings.
1204	Cannot add value *value*.	Use the NerveCenter Administrator to check the configuration settings. Invalid key is likely.
1205	Cannot read configuration settings value in MapSubNets key.	Use the NerveCenter Administrator to check the configuration settings. Invalid key is likely.

TABLE A-14. Server Manager Error Messages (Continued)

#	Error	Resolution
1206	Invalid configuration settings Entry for the value Method in the Platform key.	Only Manual and Auto are allowed. Check for case.
1207	Cannot read configuration settings value: *value*	Use the NerveCenter Administrator to check the configuration settings. Invalid key is likely.
1208	Cannot write configuration settings Value: *value*	Use the NerveCenter Administrator to check the configuration settings. Invalid key is likely.
1210	Cannot find License key in configuration settings.	Use the NerveCenter Administrator to check the configuration settings. Invalid key is likely.
1300	*value* (Import behavior/database error).	Try to resolve using the message. If not, call support.
1313	Server alarm instance maximum exceeded. Please restart Server.	Restart server.
2001	The account NCServer.exe is running under does not have the advanced user right "Act as part of the operating system."	Use User Manager to give advanced user right to the group or user that NCServer is running under. You will have to stop and restart NCServer.exe
2002	The user or a group the user belongs to does not have the advanced user right "Logon as a batch job."	Use User Manager to give advanced user right to the group or user.
2003	The user ID *value* does not exist.	Type in a user ID that exists. Check User Manager.
2004	The password is incorrect for user ID *value*.	Type in a legal password for the user ID you entered
2005	License violation. Exceeded number of allowed nodes. The number of managed nodes exceeds the limits of the license.	Either unmanage some nodes or contact your authorized sales representative for an upgrade.
2006	One of the following messages: ◆ Invalid Product ID in license key. ◆ No nodes specified in license. ◆ No users specified in license. ◆ Illegal start date specified.	Check with customer support to see that hte license was generated correctly.
	Invalid License Key.	NerveCenter could not decode the license. Check for typographical errors in the key or call support to get the key validated and/or replaced.
	License will expire in less than 14 days.	Your NerveCenter evaluation license will expire within 14 days. Contact sales or support to extend the license.

TABLE A-14. Server Manager Error Messages (Continued)

#	Error	Resolution
	License has expired.	Your NerveCenter evaluation license has expired. Contact sales or support to get the license extended.
2007	The ncadmins, ncusers not defined on the server machine and the user does not have root permissions.	Log in as root to connect to the Server. If you cannot log in as root, do one of the following: ◆ If your system uses NIS, define the groups ncadmins and ncusers on the NIS server machine, in the /etc/group file, and rebuild the NIS database. ◆ If you system does not use NIS, define the two groups in the /etc/group file of the machine where the Server is running.
2008	User does not have either administrator or user permissions.	Log in as root to connect to the Server. If you cannot log in as root, do one of the following: ◆ If your system uses NIS, include your user ID in either the ncadmins or ncusers group on the NIS server machine, in the /etc/group file, and rebuild the NIS database. ◆ If your system does not use NIS, include your user ID in either the ncadmins or ncusers group on the machine where the Server is running.
3001	Request to delete the node *value* failed because the node doesn't exist.	N/A
3002	Failed to find socket in server's map. Line: *value*.	
3003	Exiting due to a SIGTERM signal.	
3004	Primary thread initialization successful.	

Trap Manager Error Messages

Following is a list of Trap Manager error messages.

TABLE A-15. Trap Manager Error Messages

#	Error	Resolution
1	Error in TrapManagerWnd::Initialize - failed to create GetHostByAddr thread.	
2	Error in TrapManagerWnd::LaunchTrapper - failed to create trapper process.	
3	Error in TrapManagerWnd::CreateCheckTrapperThread - failed to create new thread.	
5	Error in TrapManagerWnd::InitializeMSTrapService - failed to get proc address.	
6	Error in TrapManagerWnd::InitializeMSTrapService - error from SnmpMgrTrapListen (last error).	
7	Error in TrapManagerWnd::InitializeMSTrapService - failed to create trap listen thread.	
8	Error in TrapManagerWnd::Initialize - Failed to create trap stream socket.	
9	Error in TrapManagerWnd::Initialize - Failed to listen on trap stream socket.	
10	Error in TrapManagerWnd::OnTraceTraps - Failed to create trace file for traps.	
1001	CTrapManagerWnd::OnTrapExist - gethostbyname from trap data with snmptrap failed for *value*.	
1002	Error in trap service or trap service down.	Check Windows SNMP service.
1003	CTrapManagerWnd::OnInvalidSignature - Error in receiving data on NC socket.	Check for consistent version numbers of trapper and NerveCenter executables.
1004	Expected MSTRAP or OVTRAP in NerveCenter configuration settings.	Reinstall and choose the appropriate platform integration.
2001	MS Trap service threw exception in GetTrap.	Make sure you aren't making SNMP get requests to port 162.
2002	Error processing trap data.	Make sure you aren't making SNMP get requests to port 162.
3001	Trap Manager Initialization successfully finished.	

TABLE A-15. Trap Manager Error Messages (Continued)

#	Error	Resolution
3002	Check Trapper—Trapper process died. restarting Trapper.	

NerveCenter installation Error Messages (UNIX)

Following is a list of NerveCenter installation error messages.

TABLE A-16. NerveCenter Installation Error Messages (UNIX)

Error	Resolution
Space under *dirname* is INSUFFICIENT to install LogMatrix NerveCenter	Free up space in the file system by removing files, or choose another place for installation.
The directory *dirname* must reside on a local disk	The directory you specified for NerveCenter installation is on a disk that is not on the local file system. Pick a new directory or re-mount the disk.
Write permission is required by root for *dirname* directory	The directory you specified for NerveCenter installation does not have write permission for root. Choose another directory or change the permissions.
Please create the desired destination directory for NerveCenter and re-run the installation script	The directory you specified for NerveCenter installation does not exist. Choose another directory or create the original.
Invalid mount point	The installation script could not find the CD-ROM drive and prompted you for its location. The path you specified was not valid. Verify that the drive exists, is mounted, and is configured correctly.
ProcessName is running on the system. Please exit from (or kill) *processName* process.	The installation script found that the nervectr or ovw process was running. Exit from or kill the process and re-run the installation script.
These processes must be stopped before NerveCenter can be installed. Please kill these processes and re-run the installation script.	The installation script found processes that need to be killed before installation, you were prompted to stop them, and you said no. You must manually exit from or kill the processes and re-run the installation script.
hostname is not a valid host name	The host that you provided to the script is not a valid host. Check the name of the host (capitalization, spelling, and so on) and try again.
I don’t know how to install on this architecture	Installation is supported for Solaris. The script issues this message if attempting to install on an architecture that is not in this set.
Can’t cd to *installation_path*/userfiles	Make sure the directory exists and has appropriate permissions.

TABLE A-16. NerveCenter Installation Error Messages (UNIX) (Continued)

Error	Resolution
Can’t open *hostname*.conf	The script couldn’t create the file or couldn’t open an existing configuration file. Check *installation_path*/userfiles to make sure that root has permission to write in this directory, that *hostname*.conf has read permission set, if it exists, and that localhost.conf exists and has read permission set.
Can’t create *hostname*.ncdb Can’t create *hostname*.node	The script was attempting to create the indicated file by copying data from another file. Check *installation_path*/userfiles to make sure that root has permission to write in this directory, and that localhost.*ext* exists and has read permission set.
Can’t open /etc/rc Couldn’t re-create /etc/rc Couldn’t modify /etc/rc	The script couldn’t modify /etc/rc to call the NerveCenter rc script. Edit the file and add a line that executes *installation_path*/bin/rc.openservice. There’s no need to rerun the installation script after this correction.
Can’t append to /etc/rc.local	The script couldn’t modify /etc/rc.local to call the NerveCenter rc script. Edit the file and add a line that executes *installation_path*/bin/rc.openservice. There’s no need to rerun the installation script after this correction.
Can’t create /etc/rc2.d/K94ncservice on Solaris	The script couldn’t create the NerveCenter rc script /etc/rc2.d/K94ncservice on Solaris. Copy *installation_path*/bin/rc.openservice to /etc/rc2.d//K94ncservice. There’s no need to rerun the installation script after this correction.
An error occurred in trying to contact the Server “*hostname*”. As a result, the information that you have specified cannot be used to complete this NIS update. Unable to modify *filename*. It doesn’t exist! Unable to modify *filename*. File size is 0!	The script was attempting to update system services and failed. Correct the specific error (perhaps the host name or file name was entered incorrectly) and rerun the script. If the error isn’t easily corrected, you can edit /etc/services yourself. Make sure that the following lines are included in the file: SNMP 161/udp SNMP-trap 162/udp If you’re running NIS, be sure to make these changes on the NIS server, change to the NIS directory, and run make services.

OpenView Configuration Error Messages (UNIX)

Following is a list of OpenView configuration error messages.

TABLE A-17. OpenView Configuration Error Messages (UNIX)

Error	Resolution
OpenView configuration was not entirely successful. You need to double-check the steps that failed above.	This message is displayed if any part of the OpenView configuration failed. Scroll through the script output, looking for messages that include *FAILED*. Immediately following such a line will be the error messages that resulted from the part of the script that failed.
Installing registration...FAILED	The script was attempting to copy a file into *NNM_dir*/registration/C, where *NNM_dir* is the location of your OpenView installation. Make sure that this directory exists and that root has write permission for it.
Couldn't create *NNM_dir*/help/C/ncapp	The script was attempting to create the directory *NNM_dir*/help/C/ncapp, where *NNM_dir* is the location of your OpenView installation. Make sure that help/C exists and that root has write permission for it.
Installing Help...FAILED	The script was attempting to copy files into *Network Node Manager_dir*/help/C/ncapp. Make sure the directory exists and that root has write permission for it. If you got the previous error message, you will also receive this one.
Installing Fields...FAILED	The script was attempting to copy a file into *NNM_dir*/fields/C. Make sure the directory exists and that root has write permission for it.
Installing Symbols...FAILED	The script was attempting to copy a file into *NNM_dir*/symbols/C. Make sure the directory exists and that root has write permission for it.
Installing Bitmaps...FAILED	The script was attempting to copy files into *NNM_dir*/bitmaps/C. Make sure the directory exists and that root has write permission for it.
Notifying <<OpenView...>> FAILED	The script was attempting to execute ovw. Make sure that root has appropriate permissions for ovw and that you have run ovstartup on this computer.
Installing Events...FAILED	The script was attempting to execute xnmevents. Make sure that root has appropriate permissions for xnmevents and that xnmtrap is not running on this computer.

A Error Messages

Troubleshooting NerveCenter

B

NerveCenter's complexity means that users will inevitably run into difficulties at some point. Before calling NerveCenter Technical Support, there are several first steps you can take. Besides checking the obvious—the NerveCenter Server is running, the license has not expired, etc.—you may want to look at information found in this appendix.

This appendix includes the following sections:

Section	Description
Common problems on page 260	Lists some of the most common problems NerveCenter users face.
Using Trace Counters to Troubleshoot NerveCenter on page 262	Explains how advanced NerveCenter users can use trace counters to detect problems.
Troubleshooting ASN.1 files on page 271	Explains how NerveCenter users can edit MIB files that fail to compile.

Common problems

The following list includes some of the more common problems users face when administering NerveCenter.

The NerveCenter Server

- *UNIX will not start the NerveCenter Server on page 40*.
- *An alarm causes the NerveCenter Server to crash every time I start it on page 40*.
- *The NerveCenter Server does not start on Windows on page 40*.
- *I need to get a better idea of what the NerveCenter Server is doing on Windows. Can I temporarily run it as a process? on page 40*
- *UNIX will not start the NerveCenter Administrator on page 50*.
- *While trying to connect to a NerveCenter Server I get the message: The server did not respond on page 50*.
- *I misspelled a server name while trying to connect and now the misspelled name appears in the NerveCenter Administrator Server Name list on page 50*..
- *I need to make the same changes to several NerveCenter Servers on page 71*.
- *Importing imputil.ini caused unwanted changes to a NerveCenter Server on page 71*.
- *I cannot log in to a UNIX NerveCenter Server. on page 71*

The NerveCenter Node List

- *NerveCenter is not filtering a node by a capability on page 121*.
- *After setting an IP filter, a node that should be masked out still appears in the node list on page 121*.
- *Even though I have enabled Process Traps From Unknown Nodes, NerveCenter does not update its node list when it receives a trap from an unknown node on page 121*.
- *NerveCenter does not recognize my network management platform as a valid source of node data on page 121*.
- *NerveCenter is not receiving node data from my network management platform on page 122*.
- *The IPSweep behavior model will not work on page 122*.

- *The NerveCenter node list contains two nodes with the same address but different names on page 122*.
- *NerveCenter deletes a node I added manually on page 122*.
- *NerveCenter adds a node I deleted manually on page 123*.
- *I'm seeing several errors recorded in the application event log window stating that ipsweep.exe is not running on page 123*.

Alarm Actions

- *NerveCenter is not sending SMTP mail notifications on page 194*.
- *Microsoft mail notifications are not being sent as NerveCenter on page 194*.
- *NerveCenter is not dialing pages correctly on page 194*.
- *The logs created by my behavior models are using too much disk space on page 194*.

The NerveCenter Database

- *The server shut down gracefully, but event log said that the database was not found. Where did the updates to the database go? on page 224*
- *Running SerializeDB from the command line keeps failing on page 224*.
- *SerializeDB won't work from the application or the command line on page 224*.
- *The data in the database is not what was imported from the serialized file on page 224*.

Management Information Bases (MIBs)

- *A MIB won't compile on page 233*.
- *A MIB won't load on page 234*.

Using Trace Counters to Troubleshoot NerveCenter

A useful tool for troubleshooting NerveCenter is the trace counters feature. Trace counters enable NerveCenter to keep track of its own activities, such as the number of SNMP traps that have been received, polls requested, Informs sent, and quite a few other operations. The cumulative count for an operation reflects total activity for either of the following durations:

- For as long as the NerveCenter Server has been running
- If you have reset a counter, since the reset was performed.

Trace counters can help you troubleshoot communication or behavior model problems.For example, if Informs are not being received by your network management platform for a particular network condition, you might trace a backward path starting with the Inform Action counters to find out where the problem arises. If informs are being sent, the problem is not with your alarm, polls, or masks. If no Informs are requested or sent, you might examine whether triggers are being issued or whether SNMP traps are being captured in trap masks by examining the counters in the Trigger and Trap pages.

If you are having difficulties with NerveCenter, a LogMatrix Technical Support representative may provide further instructions for using the Trace counters, including enabling trace logging.

NOTE
NerveCenter's trace logging feature should only be used as directed by Technical Support. Trace logging quickly consumes large blocks of memory and could impact Server performance.

Even though the data presented by the trace counters are read-only, you can reset a counter. See *Using Trace Counters on page 263* for details about viewing, resetting, and refreshing counters.

The following sections describe the meaning of each counter:

- *Alarm Action Counters on page 265*
- *Database Counters on page 265*
- *Inform Counters on page 266*
- *Inform NerveCenter Counters on page 266*
- *Log to Database Counters on page 267*
- *Log to File Counters on page 267*
- *Server Counters on page 268*
- *Node Source Counters on page 268*

- *Poll Counters on page 269*
- *Trap Counters on page 270*
- *Trigger Counters on page 270*

Using Trace Counters

This section describes how to access and use the trace counters.

TO VIEW NerveCenter TRACE COUNTERS

1. From the Server menu, choose **Trace Counts**.

   The Trace Counters window is displayed.

2. Select one of the tabs.

   The corresponding set of counters is displayed.

3. Select the **Refresh** button to update all counters.

Once you have the counters displayed, you can reset one or more counters as described in *Resetting Counters on page 263*. You may also periodically need to update all counters as described in *Refreshing the Trace Counter Display on page 264*.

Resetting Counters

When using NerveCenter Trace Counters to troubleshoot a problem, you may want to reset one or more counters. For example, if Informs are not being received by your network management platform for a particular network condition, you can isolate the problem by turning off alarms not related to the condition, resetting the related counters, and then tracing the activity for the alarm you want to monitor.

TO RESET NerveCenter TRACE COUNTERS

1. In the Trace Counters window, select the page containing the counters you want to reset.
2. Do either of the following:
   - Select the **Reset** checkbox next to the counter or counters you want to reset, and then select the **Reset** button. The counters you checked are reset.
   - To reset all counters in a page, select the **Reset All** button.

Refreshing the Trace Counter Display

When you open the Trace Counter window, you see the current values for all counters. Once opened, however, the counter information is updated only upon command. To update counter values, you must refresh the view manually.

TO UPDATE NERVECENTER TRACE COUNTERS ONCE THE WINDOW IS OPENED

- Select the **Refresh** button.

  All counters are updated.

Table B-1 provides a reference for all procedures related to the trace counters. Each of these procedures are performed within a NerveCenter Administrator after it is connected to a NerveCenter Server

TABLE B-1. Procedures Related to Trace Counters

Task	Procedure
To view the trace counters...	From the **Server** menu, choose **Trace Counts**.
To view a specific counter...	Within the **Trace Counters** window, select the appropriate tab.
To reset one or more counters...	After selecting each of the relevant **Reset** checkboxes, select **Reset**.
To reset all the counters in a page...	Select **Reset All**.
To update all the counters...	Select **Refresh**.

Alarm Action Counters

The Action tab of the Trace Counters window provides read-only information about the current NerveCenter Server’s automated actions. See *Using Trace Counters to Troubleshoot NerveCenter on page 262* for more information about using the counters.

TABLE B-2. Alarm Action Counters

Group	Field	Description
Specific action	Action Type	Select the type of action you want to monitor.
	Requested	The number of specified actions that have been requested.
	Pending	The number of specified actions that are pending.
	Completed	The number of specified actions that are completed.
Totals for All Actions	Requested	The total number of all actions that NerveCenter has requested (as opposed to the number of actions for the selected action type).
	Pending	The total number of all actions that are currently pending.
	Completed	The total number of all actions that have been completed.

Database Counters

The Database tab of the Trace Counters window provides read-only information about the current NerveCenter Server’s automated actions. See *Using Trace Counters to Troubleshoot NerveCenter on page 262* for more information about using the counters.

TABLE B-3. Database Counters

Group	Field	Description
DB Operations	Successful	The number of NerveCenter operations that have been written to the database.
	Pending	The number of database entries waiting to be written.
DB Connections	Lost	The number of lost connections with the database.
	Restored	The number of lost connections with the database that were later restored.

Inform Counters

The Inform tab of the Trace Counters window provides read-only information about the current NerveCenter Server's automated actions. See *Using Trace Counters to Troubleshoot NerveCenter on page 262* for more information about using the counters.

TABLE B-4. Inform Counters

Group	Field	Description
OVPA	Requested	The number of informs requested to be sent to the NerveCenter OpenView Platform Adapter.
	Sent	The number of informs that have been sent to the NerveCenter OpenView Platform Adapter.
	Pending	The number of informs saved in the queue ad waiting to be sent to NerveCenter's OpenView Platform Adapter.
Universal PA	Requested	The number of informs requested to be sent to NerveCenter's Universal Platform Adapter.
	Sent	The number of informs that have been sent to NerveCenter's Universal Platform Adapter.

Inform NerveCenter Counters

The Inform NC tab of the Trace Counters window provides read-only information about the current NerveCenter Server's automated actions. See *Using Trace Counters to Troubleshoot NerveCenter on page 262* for more information about using the counters.

TABLE B-5. Inform NerveCenter Counters

Field	Description
Received	The number of informs the current NerveCenter Server has received from other NerveCenter Servers.
Processed	The number of informs the current NerveCenter Server has received and processed from other NerveCenter Servers.

Log to Database Counters

The Log DB tab of the Trace Counters window provides read-only information about the current NerveCenter Server’s automated actions. See *Using Trace Counters to Troubleshoot NerveCenter on page 262* for more information about using the counters.

TABLE B-6. Log to Database Counters

Field	Description
Requested	The number of Log to Database actions requested.
Pending	The number of Log to Database actions that are still pending.
Lost	The number of Log to Database actions that did not successfully log and are considered to be lost.
Deleted	The number of Log to Database actions that did not successfully log and have been removed from the log queue.

Log to File Counters

The Log File tab of the Trace Counters window provides read-only information about the current NerveCenter Server’s automated actions. See *Using Trace Counters to Troubleshoot NerveCenter on page 262* for more information about using the counters.

TABLE B-7. Log to File Counters

Field	Description
Requested	The number of Log to File actions requested.
Pending	The number of Log to File actions that are still pending.
Lost	The number of Log to File actions that did not successfully log and are considered to be lost.
Deleted	The number of Log to File actions that have been removed from the log file.

Server Counters

The Server tab of the Trace Counters window provides read-only information about the current NerveCenter Server's automated actions. See *Using Trace Counters to Troubleshoot NerveCenter on page 262* for more information about using the counters.

TABLE B-8. Server Counters

Field	Description
Alarms transitioned	The number of alarm instances that the current NerveCenter Server has transitioned.
Messages to client	The number of messages the current NerveCenter Server has sent to a connected NerveCenter Client.
Client connections	The number of NerveCenter Clients that have connected to the current NerveCenter Server.
Web connections	The number of NerveCenter Web Clients that have connected to the current NerveCenter Server.

Node Source Counters

The Node Source tab of the Trace Counters window provides read-only information about the current NerveCenter Server's automated actions. See *Using Trace Counters to Troubleshoot NerveCenter on page 262* for more information about using the counters.

TABLE B-9. Node Source Counters

Field	Description
Nodes received	The total number of nodes that NerveCenter has received from a node source and stored in its database received during the last resync
Nodes updated	The total number of updates that NerveCenter has received from a node source during the last resync.
New nodes created	The number of nodes added to the NerveCenter node list when nodes were added to the network management platform.
Nodes auto deleted	The number of nodes removed from the NerveCenter node list.

Poll Counters

The Poll tab of the Trace Counters window provides read-only information about the current NerveCenter Server’s automated actions. See *Using Trace Counters to Troubleshoot NerveCenter on page 262* for more information about using the counters.

TABLE B-10. Poll Counters

Group	Field	Description
SNMP	Requests	The number of SNMP polls requested.
	Pending	The number of polls waiting for an SNMP reply poll request.
	Responses Received	The number of SNMP responses that have been received from SNMP requests.
	Error Received	The number of SNMP and ICMP errors resulting from SNMP requests.
	Polls Retried	The number of SNMP polls that were reissued.
	Polls Timed Out	The number of SNMP polls that did not receive a response and therefore timed out.
ICMP	Requests	The number of ICMP polls requested.
	Pending	The number of polls waiting for an ICMP reply to the poll request.
	Responses	The number of ICMP responses that have been received from ICMP requests.
	Error Received	The number of ICMP errors resulting from ICMP requests: The following types of messages indicate errors: ICMP_UNREACH, ICMP_SOURCEQUENCH, ICMP_REDIRECT, ICMP_TIMXCEED, ICMP_PARAMPROB.
	Polls Retried	The number of ICMP polls that were reissued.
	Polls Timed Out	The number of ICMP polls that did not receive a response and therefore timed out.

Trap Counters

The Trap tab of the Trace Counters window provides read-only information about the current NerveCenter Server's automated actions. See *Using Trace Counters to Troubleshoot NerveCenter on page 262* for more information about using the counters.

TABLE B-11. Trap Counters

Field	Description
Received	The number of SNMP traps that the current NerveCenter Server has received and processed.
Failed	The number of SNMP traps that the current NerveCenter Server has received but was unable to process.
Ignored	The number of SNMP traps that the current NerveCenter Server has received but ignored. NerveCenter ignores traps if NerveCenter is configured not to process traps from unknown nodes and the node sending the trap is outside NerveCenter's specified IP subnet filter range.

Trigger Counters

The Trigger tab of the Trace Counters window provides read-only information about the current NerveCenter Server's automated actions. See *Using Trace Counters to Troubleshoot NerveCenter on page 262* for more information about using the counters.

TABLE B-12. Trigger Counters

Field	Description
Masks	The number of triggers that NerveCenter has fired from trap masks.
Actions	The number of triggers that NerveCenter has fired from alarm actions.
Polls	The number of triggers that NerveCenter has fired from polls.
Total	The total number of triggers that NerveCenter has fired.

Troubleshooting ASN.1 files

Many mibcomp error messages are caused by problems with ASN.1 MIB files. This section describes how to make sure your ASN.1 files can be processed correctly by the mibcomp utility and how to correct common ASN.1 errors. If you receive mibcomp error messages, check your ASN.1 files for the following possible problems:

- Improper characters
- Improper use of period, underscore, and hyphen
- Improper capitalization
- Improper table construction
- Duplicate type definitions
- Object identifier format
- Ignored **IMPORT** definitions

Use of Characters

Blank spaces, carriage returns (^M), and line feeds (^J) are considered white space, used only as separators.

The following are the only characters you can use in ASN.1:

```
abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789
:=,{}<>.()[]-’”;|
```

Except in a comment, you cannot use the following characters:

```
!@#$%^&*_+\~‘?/
```

Note the following exceptions:

- You can use any character in a comment.
- An object name can have only alphabetic and numeric characters and hyphens (but never two hyphens in a row), and cannot end with a hyphen.

Period, Underscore, and Hyphen

The only nonalphanumeric character you can use in a type or value is the hyphen (-). Do not use a period (.) or an underscore (_).

Two consecutive hyphens (--) begin comments. Two consecutive hyphens or the end of a line can end comments. When you end a comment with a second pair of hyphens, mibcomp tries to convert the material between the second pair of hyphens and the end of the line. For example, given the following line:

```
-- This is a comment -- This is not a comment.
```

The mibcomp utility tries to convert “This is not a comment.” Don’t use two consecutive hyphens in an object name.

Capitalization

Key words and predefined types must contain only uppercase letters and must not be reserved character sequences. Reserved character sequences are listed below:

TABLE B-13. Reserved character sequences

ABSENT	DEFINED	INTEGER	REAL
ANY	DEFINITIONS	MAX	SEQUENCE
APPLICATION	END	MIN	SET
BEGIN	ENUMERATED	MINUS-INFINITY	SIZE
BIT	EXPLICIT	NULL	STRING
BOOLEAN	EXPORTS	OBJECT	TAGS
BY	EXTERNAL	OCTET	TRUE
CHOICE	FALSE	OF	UNIVERSAL
COMPONENT	FROM	OPTIONAL	WITH
COMPONENTS	IDENTIFIER	PLUS-INFINITY	
correct	IMPLICIT	PRESENT	
DEFAULT	INCLUDES	PRIVATE	

Type references and type declarations must begin with an uppercase letter. Enumeration names, value names, and object identifiers must begin with a lowercase letter.

Table Construction

If a table isn’t being displayed properly, it might be missing an **INDEX** clause or a **SEQUENCE** definition. An **INDEX** clause looks like this:

```
INDEX(instance)
   ::= (parent)
```

A **SEQUENCE** definition for each table in the ASN.1 file is necessary for compliance with RFC1212. For example, a MIB might have:

```
commonB   OBJECT IDENTIFIER ::= { rr2board 1 }
boardIndex   OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
          “The slot number of this board.”
::= { commonB 1 }

.
.
.
```

etc.

This prevents the mibcomp utility from producing usable output because it cannot detect that boardIndex is an attribute of a tabular object. One way to fix this is to add extra material between commonB and the first attribute, as follows:

```
commonB OBJECT IDENTIFIER ::= { rr2board 1 }
-- -- Modified for 1212 compliance --
commonB OBJECT-TYPE
SYNTAX CommonB
ACCESS read-write
STATUS mandatory
INDEX { boardIndex }
:= { rr2board 1 }
CommonB ::= SEQUENCE {
boardIndex
INTEGER,
boardName
OCTET STRING,
boardType
INTEGER,
boardTotalPorts
INTEGER,
boardStatus
INTEGER,
boardPortsOn
INTEGER }
boardIndex OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
“The slot number of this board.”
::= { commonB 1 }
.
.
```

etc.

Do not nest tables. Nesting can happen indirectly—for example, when a type that contains a table is referenced in a table without the connection being obvious. To avoid nesting, you can use an index from one table to index another.

Duplicate Type Definitions

If the mibcomp utility encounters duplicate type definitions, it uses the first definition it encounters and then displays the following message:

```
Warning(file.asn1:about line n):
Duplicate type defined: TypeName
```

It then continues processing without pausing for you to respond. Make sure the type definitions are all the same, or rename the types and use mibcomp again.

Object Identifier Format

The only object identifier format mibcomp accepts is:

```
{objname integer}
```

Any other format results in the error message:

```
Illegal Object Identifier Value
```

If your MIB contains an object identifier like:

```
open OBJECT IDENTIFIER ::= { osi 1 2}
```

you could change it to an acceptable object identifier by splitting the definition as follows:

```
open OBJECT IDENTIFIER ::= { osi 1}
open OBJECT IDENTIFIER ::= { osi 2}
```

Ignored IMPORT Definitions

The mibcomp utility ignores the **IMPORT** definition.

For example,

```
MIB-A: x = 1
MIB-B: x = 2
MIB-C: import x from MIB-B
```

In MIB-C, the utility takes x as 2 rather than as 1 if MIB-A is parsed first. Before parsing, check for **IMPORT** clauses and copy the **IMPORT** definitions directly into the MIBs containing them.

B Troubleshooting NerveCenter

Controlling NerveCenter from the Command Line

NerveCenter provides commands that you can run from a UNIX shell or a DOS Command Prompt window. NerveCenter also enables you to run a command line interface (CLI) from which you can issue single commands, open a CLI interactive session, or run a CLI script.

This appendix includes the following sections:

Section	Description
NerveCenter Command Line Interface on page 278	Describes the NerveCenter command line interface (CLI) and how to use it.
NerveCenter UNIX Shell/DOS Prompt Commands on page 282	Describes the NerveCenter commands that you can execute from a UNIX shell or a Windows command prompt.
Command Line Interface Reference on page 319	Describes the NerveCenter commands that you can execute from the command line interface (CLI).

NerveCenter Command Line Interface

You can use a command line interface (a UNIX shell or a DOS cmd window) to add, list, delete, and control nodes, alarms, polls, and masks. You can also connect to, display the status of, and disconnect from NerveCenter servers — either manually or from a script.

The following sections describe how to control NerveCenter from the command line:

- *Entering a Single Command in a UNIX Shell or a DOS Prompt on page 278*
- *Entering Multiple Commands in an Interactive Session on page 279*
- *Running a Script Containing One or More Commands on page 280*

Command Line Interface Reference on page 319 describes each specific command. The procedures in this appendix assume that you are in the NerveCenter bin directory or that your search path includes that directory.

Entering a Single Command in a UNIX Shell or a DOS Prompt

To issue a single command in a UNIX shell or at a DOS prompt

1. Open a UNIX shell or a DOS Command Prompt window.
2. Type

   ```
   nccmd -s ServerName -u UserName -p Password -n Port Command
   ```

   where *ServerName* is the name of the server you want to contact, *UserName* and *Password* are valid for that NerveCenter server, *Port* is the command-line-interface port (32506 by default), and *Command* is the command you want to run. Then press Enter.

   Note
   User IDs and passwords are encrypted before being sent over the network.

   If you are already connected to a server via a previous connect or select server command, the command you issue is directed to that server, and the -s switch is unnecessary.

The command is executed and informs you if an error was encountered. The command syntax is specified in *Command Line Interface Reference on page 319*.

In the following example, a user named John lists all the polls that are suppressed. For this example, the server is named sales, John's user ID is johndoe, and his password is anytown.

```
nccmd -s sales -u johndoe -p anytown -n 32506 list poll -S yes
```

Entering Multiple Commands in an Interactive Session

To open an interactive session and enter multiple commands

1. Open a UNIX shell or a DOS Command Prompt window.
2. Do one of the following:
   - To open a session without connecting directly to a server, type **nccmd** and press Enter.

     The nccmd command prompt (`nc>`) appears.
   - To open an interactive session and connect directly to a server, type

     `nccmd -s` *`ServerName`* `-u` *`UserName`* `-p` *`Password`* `-n` *`Port`*

     where *ServerName* is the name of the NerveCenter server you want to contact, *UserName* and *Password* are a valid for that NerveCenter server, and *Port* is the server's command-line-interface port (32506 by default). Then press Enter.

     The nccmd command prompt (`nc>`) appears and you are connected to the server.

     All subsequent commands are sent to that server until you connect to an additional server or issue a select server command.
3. Type the commands you want to run, pressing Enter after each one.

   The command syntax is specified in *Command Line Interface Reference on page 319*.

   Each time you press Enter, you are returned to the nccmd command prompt. If any command returns an error, nccmd quits, returns you to the shell prompt or the Command Prompt, and displays the error.
4. To exit the interactive session, type **quit** at the prompt and press Enter.

   You are returned to a shell or command Prompt.

In the following example, a user named Kim opens an interactive session, connects directly to the server, deletes a node, disconnects from the server, and quits the session. For this example, the server is named sales, Kim's user ID is kimz, her password is NoMuZak, and the name of the node she is deleting is N0128.

```
C:\>nccmd -s sales -u kimz -p NoMuZak -n 32506
nc>delete node -n N0128
nc>disconnect
nc>quit
```

Running a Script Containing One or More Commands

TO RUN A SCRIPT THAT CONTAINS ONE OR MORE COMMANDS

1. Create a text file that contains the commands in the order you would enter them as if you were in an interactive session.

   The command syntax is specified in *Command Line Interface Reference on page 319*.

2. Open a DOS Command Prompt window or a UNIX shell.

3. Do one of the following:

   - To connect to a NerveCenter server and have the server execute the commands in your file, enter a command of the following form:

     ```
     nccmd -s ServerName -u UserName -p Password -n Port -f Filename
     ```

     ServerName is the name of the NerveCenter server you want to contact, *UserName* and *Password* are a valid user name and password for that NerveCenter server, *Port* is the command-line-interface port for the server (32506 by default), and *Filename* is the name of the file containing the commands you want to run.

     NOTE
     If *Filename* is a file name or path name that contains spaces, the name must be enclosed in quotation marks, for example, “C:\Program Files\LogMatrix\NerveCenter\Bin\Cmd.nc”.

   - To prevent having to connect to the server explicitly each time you want to run your script, you can edit your script so that its first line is a connection command. This command must have the form:

     ```
     connect -s ServerName -u UserName -p Password -n Port
     ```

     Once you’ve performed this step, you can type **nccmd -f** *Filename* from the DOS or a shell prompt to connect to the server and execute the commands in your script.

     If any command returns an error, nccmd quits, displays the error, and returns you to the Command Prompt or UNIX prompt.

In the following example, a user named Juan writes a script that connects to a server, turns on an alarm named Monitor, and disconnects from the server. For this example, the server is named sales, Juan's user ID is juanr, and his password is CaFFieNe. The script is named SetAlarm and it looks like this:

```
connect -s sales -u juanr -p CaFFieNe -n 32506
set alarm -e on -w -a Monitor
disconnect -s sales
```

Juan runs the script by entering the following command at the command prompt:

```
>nccmd -f SetAlarm
```

NerveCenter UNIX Shell/DOS Prompt Commands

There are certain NerveCenter commands that you run from a UNIX shell or a Windows command prompt. This section lists each of these commands in alphabetical order:

- *client on page 283*
- *dbwizard on page 283*
- *importutil on page 284*
- *installdb on page 293*
- *ipsweep on page 294*
- *mibcomp on page 296*
- *mibTool on page 298*
- *ncadmin on page 299*
- *nccmd on page 300*
- *ncserver on page 301*
- *ncstart on page 303*
- *ncstop on page 304*
- *ovpa on page 306*
- *paserver on page 310*
- *serializedb on page 313*
- *trapgen on page 314*

NOTE

This appendix assumes that you are in the NerveCenter bin directory or that your search path includes that directory. NerveCenter installation includes this directory in your search path by default.

client

NAME

client

SYNOPSIS

```
client
```

DESCRIPTION

Executes the NerveCenter Client application. For information about how to use NerveCenter Client, refer to *Designing and Managing Behavior Models* and *Monitoring Your Network*.

USAGE

If you receive the error message client: Command not found on a UNIX system, NerveCenter has not been installed in the default location (/opt/OSInc). You must either navigate to the NerveCenter bin directory before entering the command, or use a fully-qualified pathname.

NOTE

On UNIX, before you can run NerveCenter, you must first set the necessary environment variables by running the appropriate ncenv shell script. For more information see *Running the NerveCenter Server on UNIX on page 38*.

SEE ALSO

ncstop on page 304

dbwizard

NAME

dbwizard

SYNOPSIS

```
dbwizard
```

DESCRIPTION

A wizard that helps you to create a NerveCenter Microsoft Access or SQL Server database.

USAGE

InstallDB is the command-line version of DBWizard. If you are installing NerveCenter on many computers, you can use the wizard to set up the first database, save your configuration settings, and use them with the command-line version of DBWizard to save time on subsequent setup routines.

SEE ALSO

installdb on page 293

importutil

NAME

importutil

SYNOPSIS

```
importutil imputil.ini
```

DESCRIPTION

Enables you to reconfigure a setting on more than one NerveCenter Server at a time by changing one file and importing it to all the relevant servers.

OPTIONS

`imputil.ini`

File that contains NerveCenter Server settings that you import to all NerveCenter Servers. imputil.ini resides in the /userfiles directory .

USAGE

imputil.ini is made of a number of sections that include a section header and keys. Before making any changes, create a backup copy of imputil.ini.

CAUTION
You will not be able to restore the original imputil.ini after making changes to the file, unless you first make a backup copy.

Delete all but the relevant sections to be changed.

All sections in the file are optional. If you remove a section, including the section header and keys, ImportUtil does not change or delete values in the NerveCenter configuration settings for that key.

For example, if you are changing a value found only in the section [CONFIG_SERVER], you delete all sections except the section header and the values in the [CONFIG_SERVER] section. ImportUtil will only change the values pertaining to that section.

Any new values left in imputil.ini will overwrite the old values. To avoid having placeholders overwrite legitimate values, delete any unnecessary keys before running ImportUtil.

For example, if within the [CONFIG_SERVER] section you only want to change the value of the key InformNCListenPort, delete all but the following:

```
[CONFIG_SERVER]
InformNCListenPort = port
```

CAUTION

If you are configuring either [CONFIG_PLATFORM_NETNODENOTIFY] or [CONFIG_PLATFORM_MAPSUBNETS], you need to include all values, including old values. ImportUtil deletes values from the NerveCenter configuration settings that are not included in these sections. Please read the comments before each section in this file for more information.

Change the values by replacing the placeholders after the equal sign (=) with valid values. Unless otherwise noted, you may not leave the value after a key blank.

NOTE

You must either be in the same directory as the imputil.ini file or include the full pathname of the imputil.ini file.

NerveCenter notifies you upon successful completion of the reconfiguration. For more information about each section in imputil.ini, refer to *Table C-1*.

TABLE C-1. inputil.ini Value Descriptions

imputil.ini value pair	Description
[CONFIG_PLATFORM]	
This section configures integrations with network management platforms. To remove a value, leave the value after the equal sign (=) blank. For example: MapHost=	
Note: Except for OpCHost, the cross-references in this section refer to integrating with HP OpenView. You may need to review the chapter in *Integrating NerveCenter with a Network Management Platform* appropriate for your network management platform.	
MapHost = host/port/resyncparentrate	These are the hostname, port, and resync Parent rate defined on the Node source tab. For more details about this value, see *Identifying Parent-Child Relationships in Integrating NerveCenter with a Network Management Platform*
Method = Auto | Manual	When the method is set to Automatic, NerveCenter ignores your address filters. It looks at the IP address and mask for each network interface card on that server and uses them to calculate the subnet address and mask. Enter Manual if you want NerveCenter to use your address filters as entered. For details see *Filtering Nodes by IP Address on page 99*
WantedCapabilities = cap1,cap2,...	You can monitor nodes with particular capabilities, which your network management system typically assigns to a node to determine the applicable activities such as isRouter, isHub, and isIP. Filtering by capabilities is available only when your platform has assigned specific capabilities to a node. For more details about this value, see *Filtering Using Node Capabilities on page 97*.
SystemObjectIds = id1,id2,...	NerveCenter allows you to monitor managed nodes according to their particular system object identifiers (OIDs). For more details about this value, see *Filtering Using a Node System Object Identifier on page 98*.
MaxQueueDepth = maximum queue depth	The number of Informs NerveCenter keeps if connection with the network management platform goes down. For more details about this value, see *Configuring the Inform Queue Depth in Integrating NerveCenter with a Network Management Platform*
AckInforms = 0/1	If you want NerveCenter to keep track of informs until the platform adapter acknowledges their receipt, you must enable the inform acknowledgement feature by setting AckInforms = 1. For more details about this value, see *Saving NerveCenter Informs Until Acknowledgement in Integrating NerveCenter with a Network Management Platform*.

TABLE C-1. inputil.ini Value Descriptions (Continued)

imputil.ini value pair	Description
NoFiltersOnImport = 0/1	If you have a node list you want to import of known hosts you want to monitor, you may not want to apply the IP and Hostname filters. Enter 0 for false and 1 for true. For more details about this value, see *Enabling and Disabling IP and Hostname Filters on page 105*.
HostnameFilters = hostname1 *hostname2 . . .	Besides IP addresses, you can also filter by hostnames. For more details about this value, see *Filtering Nodes by Hostnames on page 104*.

[CONFIG_PLATFORM_MAPSUBNETS]

This section configures the IP filtering for node lists imported from a network management platform.

If the method in [CONFIG_PLATFORM] is AUTO, the entries in this section are ignored.

Caution: You must include all values from this section, including old values. ImportUtil deletes values from the NerveCenter configuration settings that are not included in these sections.

Note: If you want to apply filters to a list of nodes you are importing, first run importutil to change the filters. Then run importutil again to import the nodes.

imputil.ini value pair	Description

MapSubNet1 = Address;Mask;Exclusion list
MapSubNet2 = Address;Mask;Exclusion list
	You can filter nodes by subnet. For more details about this value, see *Filtering Nodes by IP Address on page 99*.

[CONFIG_PLATFORM_NETNODENOTIFY]

This section configures what information NerveCenter sends to the Network Managment Platform.

Caution: You must include all values from this section, including old values. ImportUtil deletes values from the NerveCenter configuration settings that are not included in these sections.

Note: The cross-references in this section refer to integrating with HP OpenView. You may need to review the chapter in *Integrating NerveCenter with a Network Management Platform* appropriate for your network managment platform.

imputil.ini value pair	Description
host/port/type = EVENT_ONLY | EVENT_AND_SYMBOL | SYMBOL_ONLY [Min Severity] [Properties]	The type of information sent to the network managment platform. For more details about these values, see *Sending NerveCenter Informs to OpenView in Integrating NerveCenter with a Network Management Platform*. **Note:** Type must be either OVPA or PA_SVR

TABLE C-1. inputil.ini Value Descriptions (Continued)

imputil.ini value pair	Description
[CONFIG_SERVER] This section configures the NerveCenter Server.	
ServerConnPort = port	The NerveCenter Server communicates with other applications, such as NerveCenter Client and NerveCenter Administrator, on a special connection port. For more details about this value, see *Configuring the NerveCenter Server Connection Port on page 74*. If you change a NerveCenter Server's communication port number, be sure all applications, such as NerveCenter Administrator, have a matching port number. See *Changing the NerveCenter Administrator Server Port on page 76* for more details.
CLIConnPort = port	The port number you want the NerveCenter Server to use for client/server communication. For more details about this value, see *Changing the NerveCenter Client Server Port on page 77*.
InformNCListenPort = port	Before a NerveCenter Server can receive an Inform, it must be configured to have a listening port number. For more details about this value, see *Configuring NerveCenter to Receive Inform Actions on page 78*.
NCMibName = MibFileName	The filename of the mib file you want to import into NerveCenter. You enter the path in [CONFIG_SERVER_PATH].
EnableDiscovery = TRUE | FALSE	If true, NerveCenter adds nodes discovered by traps to the database. For more details about this value, see *Adding Nodes Discovered from Traps on page 117*.
DiscoverNodesFromTraps = All | Filter | None	For more details about this value, see *Populating Using the IPSweep Behavior Model on page 112*.

TABLE C-1. inputil.ini Value Descriptions (Continued)

imputil.ini value pair	Description
OVTrapperPath = MSTRAP | OVTRAPD | NCNOMSTRAP | NCNOOVTRAP | ALWAYS | NCNEVER | NCALWAYS	**Windows**: ◆ MSTRAP - traps are received from Microsoft SNMP Trap Service. ◆ OVTRAPD - traps are received via ovtrapper from HP OpenView. ◆ NCNOMSTRAP - traps are received via the NerveCenter SNMP Trap agent. Use if NerveCenter is *not* co-resident with HP OpenView. ◆ NCNOOVTRAP - traps are received via the NerveCenter SNMP Trap agent. Use if NerveCenter is co-resident with HP OpenView. **UNIX**: ◆ ALWAYS - traps are received via ovtrapper from HP OpenView ◆ NCNEVER - traps are received via the NerveCenter SNMP Trap agent. Use if NerveCenter is *not* co-resident with HP OpenView. ◆ NCALWAYS - traps are received via the NerveCenter SNMP Trap agent. Use if NerveCenter is co-resident with HP OpenView. For more details about this value, see *The OpenView Platform Adapter in Integrating NerveCenter with a Network Management Platform*.
useOvTrapd = MSTRAP | OVTRAP	The NerveCenter trap source. For more details about this value, see *Managing the NerveCenter Trap Source on page 52*.
ApplyAllMasksForEachTrap = TRUE | FALSE	Set to TRUE if you want NerveCenter to process every incoming SNMP trap against all defined trap masks that are currently enabled. Selecting this option forces masks to process traps regardless of the state of their associated alarms. A mask processes traps even when its associated alarm is turned off or is not in a state that can be transitioned by the mask's trigger. There are times, however, when masks do not process traps though this option may be checked. A mask processes traps only when the following conditions are met: ◆ The mask is turned on. ◆ The incoming trap matches the generic, specific, or enterprise OID values defined in the mask.

TABLE C-1. inputil.ini Value Descriptions (Continued)

imputil.ini value pair	Description
[CONFIG_SERVER_PATH]	
Path information.	
MibDirectory = Location of Mib	The path to the mib file you want to import into NerveCenter. You enter the filename in [CONFIG_SERVER].
LogDirectory = Location of logs	The location of any log created by a Behavior Model or other component of NerveCenter. For details about this value, see *Specifying Settings for Log Management on page 191*.
[CONFIG_SERVER_PAGER]	
This section configures the paging properties.	
TouchPulse =	Enter the frequency. Default is 1200. For details about this value, see *Setting Dialing Properties for Paging Notification on Windows on page 188*.
DialOutNumber =	The number required to reach an outside line, for example, 9. For details about this value, see *Setting Dialing Properties for Paging Notification on Windows on page 188*.
PauseString =	The characters used to create a pause between dialing and sending data. For details about this value, see *Setting Dialing Properties for Paging Notification on Windows on page 188*.
[CONFIG_SERVER_SNMP]	
This section configures SNMP settings.	
SnmpRetry = No. of retries	The number of times to reissue unanswered SNMP or ICMP request polls. For more details about this value, see *Specifying SNMP Poll Intervals for NerveCenter on page 139*.
SnmpInterval = No. of Intervals	The number of seconds NerveCenter should wait for a reply to a poll before issuing another. For more details about this value, see *Specifying SNMP Poll Intervals for NerveCenter on page 139*.
DefaultPort = port no.	The port number NerveCenter uses for SNMP traps. For more details about this value, see *Specifying SNMP Ports for NerveCenter on page 81*.
[CONFIG_SERVER_SNMPV3]	
This section configures the SNMPv3 settings.	
PollUser = poll user	The SNMPv3 user name. For more details about this value, see *Configuring SNMPv3 Security Settings on page 147*.

TABLE C-1. inputil.ini Value Descriptions (Continued)

imputil.ini value pair	Description
PollContext = poll context	The SNMPv3 context. For more details about this value, see *Configuring SNMPv3 Security Settings on page 147*.
AutoClassify = on | off	Whether NerveCenter attempts to classify the SNMP format of discovered nodes. For more details about this value, see *Classifying Nodes Automatically on page 135*.
AuthPwd = unencrypted (plain text) authentication password	The Authentication password for an SNMPv3 node. For more details about this value, see *Configuring SNMPv3 Security Settings on page 147*.
PrivPwd = unencrypted (plain text) privacy password	The Privacy password for an SNMPv3 node. For more details about this value, see *Configuring SNMPv3 Security Settings on page 147*.
MaxClassifyVer = v1 | v2c | v3	If autoclassification is enabled, you can limit the level to which NerveCenter attempts to classify nodes. For more details about this value, see *Setting a Maximum Classify Value on page 137*.
MaxReqPerCycle = Max no. of requests going out of v3opmanager per second	You control SNMPv3 performance by setting a maximum number of requests per processing cycle (approximately one second) for all v3 operations including classification, SNMP test version poll, version change, authentication protocol change, security level change, and initialization requests. For more details about this value, see *Setting the Maximum SNMPv3 Requests per Cycle on page 138*.
updateKeys = T | F	To update the passwords on all managed agents, enter T. For more details about this value, see *Configuring SNMPv3 Security Settings on page 147*.
[CONFIG_SERVER_LOGS] This section configures the NerveCenter Logs.	
MaxLogEntryAge =	The number of hours you want to keep individual log entries before they are deleted from the rest of the log. For more details about this value, see *Specifying Settings for Log Management on page 191*.
MaxLogFileSize =	The size limit, in kilobytes, of the ASCII text file storing the results of the Log to File alarm action. For more details about this value, see *Specifying Settings for Log Management on page 191*.
MaxNumDBRecords =	The highest number of records in the database file storing the results of the Log to Database alarm action. For more details about this value, see *Specifying Settings for Log Management on page 191*.
LogDelPercentage =	The percentage of the log to clear when the maximum file size or the maximum number of records is reached. For more details about this value, see *Specifying Settings for Log Management on page 191*.

TABLE C-1. inputil.ini Value Descriptions (Continued)

inputil.ini value pair	Description
MaxQueueDepth =	The highest number of changes that you want queued before saving to the database. For more details about this value, see *Specifying Settings for Log Management on page 191*.
[CONFIG_SERVER_MSMAIL] This section configures Microsoft Mail.	
Profile =	The NerveCenter-designated Exchange profile to point to the machine hosting the Microsoft Exchange Server and the NerveCenter-designated mailbox created in *Configuring NerveCenter to Send Microsoft Mail on page 182*.
Password =	The password to the NerveCenter-designated mailbox created in *Configuring NerveCenter to Send Microsoft Mail on page 182*.
[CONFIG_SERVER_SMTPMAIL] This section configures SMTP mail.	
SMTPHost =	The name of the host running your SMTP mail server. For more details about this value, see *Specifying an SMTP Server for Mail Notification on page 180*.
[IMPORT_MODEL] Use this section to import a NerveCenter Behavior Model.	
File = model_path_and_filename	For more details about this value, see *Importing Node, Object, and Behavior Model Files in Designing and Managing Behavior Models*.
[IMPORT_NODE] Use this section to import a NerveCenter Node List.	
File = node_path_and_filename	For more details about this value, see *Importing Node, Object, and Behavior Model Files in Designing and Managing Behavior Models*.

installdb

NAME

installdb

SYNOPSIS

`installdb -F` *`IDBfile`* `-O {f|l|c|r} -D {s|a} [-M {v|s}] [-H]`

DESCRIPTION

Creates or modifies a NerveCenter Microsoft Access or SQL Server database.

OPTIONS

`-F` *`IDBfile`*

The path and file name for the IDB file (required).

`-O {f | l | c | r}`

Specifies the operation code (required). The choices are:

f—Full database creation and installation

l—Load serialized file

c—Create data source and connection string

r—Run SQL script

`-D {s | a}`

Specifies the database type (required). The choices are:

s—SQL

a—Access

NOTE
The SQL script can only run against a SQL database, not an Access database. If the operation parameter is r (run SQL script), then you must specify s (SQL) as the database type.

`[-M {v | s}]`

Specifies the mode of operation (optional). v is verbose, and s is silent. If **-M** is not specified, InstallDB defaults to s (silent).

```
[-H]
```

Displays usage help is in a dialog box. You can also enter simply: **installdb**

USAGE

Command line switches in Windows are case-insensitive; you can type them in any order.

If any of the required command line switches are missing and the mode is s (silent), then an error is written to the Event Log. If mode is v (verbose), then an error message is displayed.

EXAMPLE

```
InstallDB -F c:\temp\dbwizard.idb -M v -O f -D s
```

ipsweep

NAME

ipsweep

SYNOPSIS

```
ipsweep [-d] [-trace] [-f excludefile] [-m mapfile] [-w mapfile]
            [-notraps] [-t timedelay] [-ttl hops] | [-h]
```

DESCRIPTION

Discovers network devices on the subnets managed with NerveCenter. Ipsweep pings IP addresses according to filters set in the NerveCenter Administrator. If no filters are defined, ipsweep does not ping any IP addresses. If ipsweep receives a response from an IP address, ipsweep sends a SNMPv1 trap to NerveCenter. You must configure NerveCenter to automatically add nodes if a trap is received in order to update your node list. For more information about configuring NerveCenter to work with ipsweep, see *Populating Using the IPSweep Behavior Model on page 112* and *Using IPSweep Behavior Model in Designing and Managing Behavior Models*.

The NerveCenter Server must be running to use ipsweep.

OPTIONS

```
-d
```

Specifies debug mode and outputs verbose debug messages to the console.

```
-trace
```

Specifies trace mode and outputs status messages to the console.

`-f` *`excludefile`*

Specifies a file that contains a list of IP addresses for ipsweep to ignore.

`-m` *`mapfile`*

Specifies a map file that contains node information about the network.

IP Filters must still be configured in the NerveCenter Administrator's IP Filters page for each subnet that is discovered. The map file uses the subnet and mask value to match an existing pair in the IP Filters page. The Map file does not specify which subnets to discover. It just specifies the read community string used for a subnet if it is discovered.

`-w` *`mapfile`*

Writes a text file containing the read community string used for discovered subnets. You can then edit this file and use with -m.

`-notraps`

Specifies that no SNMPv1 Traps are sent. Useful with -d and -trace for debugging behavior without causing unnecessary notifications.

`-t` *`timedelay`*

ICMP timeout interval. *timedelay* is a value in milliseconds (default is 20 milliseconds).

`-ttl` *`hops`*

Specifies the time to live (TTL) value for IPSweep, where the valid value is between 0 and 255 hops. The default value for **-ttl** is: 255 hops (128 for Windows 2000).

`-h`

Provides command line help for ipsweep.

USAGE

Ipsweep can read an exclude file specified by the -f switch. This exclude file is parsed as follows:

- One IP Address can be listed on each line.
- The IP Address must be the first token in each line; there can be no leading white space.
- Any text or data following the IP Address is ignored.
- Each line cannot be more than 16 characters in length or else the entire line is ignored.
- Lines which are blank or do not contain a valid IP Address as the first token are ignored.
- An IP Address is considered valid only if it contains four octets, separated by periods (.), where each octet is in the range 0 (zero) to 255 inclusive.

- A hostname cannot be used in place of an IP Address; no lookup is performed to see whether a non-valid IP Address is actually a hostname.
- You can use comment labels, such as # or ;, to comment your exclude file, for example:

```
# Private QA Router
168.192.13.194
# Printer
168.192.1.44
```

On startup, ipsweep can read a map file specified by the **-m** switch. This map file contains a table as follows:

Default *DefaultReadCommunity*

Subnet1/Mask1 ReadCommunity1

SubnetN/MaskN ReadCommunityN

The first line should be the '**Default** *DefaultReadCommunity*' line. A single blank separates the word **Default** and the community string. Each of the lines following the Default line will be in the format specified above. These lines will be a Subnet immediately followed by a slash ("/") immediately followed by the Mask. The "Subnet/Mask" is followed by a single blank and then the community string, for example:

```
192.168.1.0/255.255.255.0 private
```

mibcomp

NAME

mibcomp

SYNOPSIS

`mibcomp [-trace] [-nocache] [-clean] {-help|-ver}` *`mib-definition-file`*

DESCRIPTION

Compiles the NerveCenter management information base (MIB) with SNMP Research's Bilingual Request and Security Subsystem (BRASS) SNMP MIB compiler utility suite.

After the first compilation, intermediary files are stored in mib\cache (Windows) or mibs/cache (UNIX), unless you use the `-nocache` argument.

OPTIONS

`mib-definition-file`

Name of the file that contains NerveCenter MIB definitions, the default is mibcomp.txt.

`-trace`

Gives a verbose output to monitor the process of compiling the *mib-definition-file*.

`-nocache`

Causes all MIB compilation to occur in $TMP/cache.

`-clean`

Resets the compilation cache, stored in mib\cache (Windows) or mibs/cache (UNIX), and performs a complete recompilation of all MIB modules. When removing entries from the mibcomp.txt file, you must run mibcomp with the -clean argument to remove the cached intermediary files.

`-help`

Displays description of arguments.

`-ver`

Displays mibcomp version information.

USAGE

By default, the NerveCenter MIB is /mibs/nervectr.mib (\mib on Windows). As with any compilation, it is always wise to make a backup copy of the nervectr.mib, mibnamemap.txt, and mibcomp.txt before running mibcomp; however, Open does provide backup files of the NerveCenter MIB and the MIB definition file shipped with NerveCenter, /mibs/nervectr-*release-number*.mib and mibcomp-*release-number*.txt, respectively.

mibcomp requires that you change to the mibs (mib on Windows) directory before compiling. As the mibcomp compiles the file, it displays a series of messages. If a problem needs your attention, the compilation fails. You must resolve the error and re-compile. The error is displayed on the last line on your screen. You will have to do this more than once if you have several errors. Errors are generally syntax problems in third-party MIBs or MIBs that do not comply with the RFC specification.

⚠ **CAUTION**

The MIB must be located on the machine hosting NerveCenter Server. Configure security on the machine so unauthorized users cannot overwrite or change the MIB.

You must reload theNerveCenter MIB after compiling it. See *Managing Management Information Bases (MIBs) on page 225* for more information.

mibTool

NAME

mibTool

SYNOPSIS

```
../bin/mibTool [-o filename] [-loc path] [-inc mib_name] -mibcomp
         filename [-namemap] [-strict] [-e filename]
```

DESCRIPTION

mibTool is a new Java-based NerveCenter MIB compiler that is installed with the Server component. mibTool produces a compiled NerveCenter MIB identical to the one produced by mibcomp in prior releases. mibTool, however, provides a marked performance improvement and has a far greater allowance for SNMP syntax variances as compared to mibcomp.

mibTool is a Java application and requires that Java 1.5 or higher is installed on the system from which it is run. NerveCenter 5.1 supports the prior MIB compiler, mibcomp, though support will be withdrawn in the next release.

OPTIONS

`-o` *`filename`*

Specifies the output filename for compiled MIB information (nervectr.mib, by default).

`-loc` *`path`*

The directory being used as a MIB repository; all MIBs in this directory are checked if an entity lookup fails.

`-inc` *`mib_name`*

Includes the specified MIB in the output file.

`-mibcomp` *`filename`*

Specifies the file that contains the MIB file paths to compile (mibcomp.txt by default).

`-namemap`

Produces the mibnamemap.txt file for use with older versions of NerveCenter.

`-strict`

Compiles in strict mode, which succeeds only if all imports are completely accurate.

`-e` *`filename`*

Writes compilation errors to the specified file.

ERROR MESSAGES

Duplicate entity detected

Multiple mib objects were detected with the same name but different OIDs. All objects under the enterprises OID will be renamed to *MIB-NAME_entityName* to allow the objects to be accessible from within NerveCenter.

Not renaming "mib object"

A duplicate entity was found but its OID was not under enterprises therefore the entity was not renamed as described in the "Duplicate entity detected" message.

Duplicate OID found

Multiple mib objects with different names were found using the same OID. The compiler will use the first name it encounters and discard the remaining duplicates.

Requested entity "mib object" imported from MIB was found in ATL-MIB instead

The compiler could not find the mib object in the file specified in the imports clause, but it was found in another mib file. This behavior only occurs when compiling without the `-strict` flag.

No such file or directory

A file specified on the command line or within the mibcomp.txt file could not be found.

ncadmin

NAME

ncadmin

SYNOPSIS

```
ncadmin
```

DESCRIPTION

Executes the NerveCenter Administrator application.

USAGE

On UNIX, if you receive an error message, it is possible the necessary environment variables have not been set. See *Running the NerveCenter Server on UNIX on page 38*.

SEE ALSO

ncserver on page 301, *ncstop on page 304*

nccmd

NAME

nccmd

SYNOPSIS

`nccmd [connect -s` *`server`* `-u` *`userID`* `[-p` *`password`*`] -n` *`port`* `[-f` *`scriptFile`* `|` *`CLIcommand`*`]] | [-?]`

DESCRIPTION

Runs the NerveCenter command line interface (CLI). You can issue single commands, open a CLI interactive session, or run a CLI script.

OPTIONS

`connect`

Connects to the specified server. Once you are connected to a server, all subsequent commands are directed to that server.

`-s` *`server`*

Name of the server to which you want to connect. Required.

`-u` *`userID`*

User ID under which you want to connect to the server. The default is the user ID specified during the previous connection or by the nccmd command line parameter, if any. Required.

`-p` *`password`*

Password for the user ID you specified. The default is the password specified during the previous connection or by the nccmd command line parameter. **-p** is required for sites that use NerveCenter security.

`-n` *`port`*

Number of the port you want to use for client/server communication. The default command-line-interface port is 32506. Required.

`-f` *`scriptfile`*

Filename of the script or batch file containing CLI commands.

NOTE
If *scriptfile* is a file or path name that contains spaces, it must be enclosed in quotes, for example, "C:\Program Files\OpenService\NerveCenter\Bin\Cmd.nc".

To avoid connecting to the server each time you run your script, you can edit your script so that its first line is a connection command. Once done, you can type **nccmd -f** *scriptfile* from a Windows cmd prompt or UNIX shell to connect to the server and execute the script. If any command returns an error, nccmd quits, displays the error, and returns you to the Windows command prompt or UNIX shell prompt.

`CLIcommand`

Any one of several command line interface commands listed in the *Command Line Interface Reference on page 319*.

`-?`

Provides command-line help for the nccmd command.

nclicreport

NAME

nclicreport

SYNOPSIS

```
nclicreport
```

DESCRIPTION

Prints license key information needed for obtaining the license file required for NerveCenter 5.1.

ncserver

NAME

ncserver

SYNOPSIS

```
ncserver [-fg | -info | -off]
```

DESCRIPTION

Executes the NerveCenter Server application. For information about how to use NerveCenter Server, refer to *Managing the NerveCenter Server on page 51*.

OPTIONS

`-fg`

Runs NerveCenter in the foreground in a command prompt. Useful when debugging NerveCenter.

`-info`

On machines installed with NerveCenter Server, displays Server version and copyright information. The Server does not have to be running to use **-info**.

`-off`

(UNIX only) Disables all alarms when you start the server. (For example, you might want to use **-off** if you create an alarm that causes undesirable results on the server.)

`-pamdebug`

(UNIX only) Causes added output to syslog for diagnosing login attempts. The messages to syslog reveal information regarding why denials are given to login attempts. This often will couple with syslog messages given off by PAM elements, giving an administrator stronger ability to resolve login issues. *Chapter 11, Managing NerveCenter Security*.

USAGE

Using **ncserver** on Windows runs NerveCenter Server as a process. To run the Server as a Windows service, use **ncstart -p**.

If you use ncstart -p, you must also start the NerveCenter SNMP service by using the following command:

```
ncsnmp start
```

You can stop the NerveCenter SNMP service by using the following command:

```
ncsnmp stop
```

If NerveCenter is not running as a UNIX daemon, you can start it manually.

Before you can run any NerveCenter application on UNIX, you must set the necessary environments by executing one of the following shell scripts:

- *installPath*/userfiles/ncenv.sh
- *installPath*/userfiles/ncenv.csh
- *installPath*/userfiles/ncenv.ksh

where *installPath* is typically //opt/OSInc

TIP

An administrator should add the lines from these files to the login scripts of all users who are going to be using NerveCenter on UNIX.

From the *installPath*/bin directory, where *installPath* is typically /opt/OSInc type the following command:

```
ncserver &
```

SEE ALSO

ncstart on page 303

ncstart

NAME

ncstart

SYNOPSIS

```
ncstart -h | -p | -s
```

DESCRIPTION

Runs the NerveCenter Server as either a process or a Windows service.

OPTIONS

NOTE

The following three options are available only on Windows machines.

`-h`

Displays command line help for ncstart.

`-p`

Runs the NerveCenter Server as a process.

`-s`

Runs the NerveCenter Server as a Windows service.

NOTE
The following three options are available only on UNIX machines.

`-fg`

Runs NerveCenter in the foreground in a command prompt. Useful when debugging.

`-info`

On machines installed with NerveCenter Server, displays Server version and copyright information. The Server does not have to be running to use **-info**.

`-off`

(UNIX only) Disables all alarms when you start the server. (For example, you might want to use **-off** if you create an alarm that causes undesirable results on the server.)

USAGE

During a NerveCenter installation, the NerveCenter Server is installed as a Windows service. During installation, the choice is given to have the NerveCenter Server service start up automatically.

If the NerveCenter Server is installed as a service but not as an automatic service, you must start it manually each time.

SEE ALSO

ncserver on page 301, *ncstop on page 304*

ncstop

NAME

ncstop

SYNOPSIS

```
ncstop -a | -All | -c | -h | -s
```

DESCRIPTION

Shuts down one or all NerveCenter applications.

OPTIONS

`-a`

Shuts down the NerveCenter Administrator application.

`-All`

Shuts down all NerveCenter currently running. -All is case-sensitive.

`-c`

Shuts down the NerveCenter Client application

`-h`

Displays command line help for ncstop.

`-s`

Shuts down the NerveCenter Server application

SEE ALSO

client on page 283, *ncadmin on page 299*, *ncserver on page 301*, *ncstart on page 303*

ovpa

NAME

ovpa

SYNOPSIS

```
ovpa [-logLevel Error | Warning | Debug | Trace ]
     [-traceOutputFile FileName ] [ -traceOutputStdout ]
     [-traceResync ] [ -traceInform ] [ -traceLoadDb ]
     [-traceParentComp ] [ -traceMapUpdates ] [ -traceConnects ]
     [-traceAll ] [ -pc ] [ -writeParentsToFile NcHostName ]
     [-heartbeat seconds ] [ -ignoreUnmanagedIntf ]
     [-initNcStatusInOV ] [ -ignoreCapability ]
     [-noResolveCommunityOnResync ] [-defGlobalCommunity filename]
     [-preferredAddrOnly]
```

DESCRIPTION

Runs the NerveCenter OpenView Platform Adapter (OVPA).

OpenView (OV) must be running before you can run OVPA. To issue an **ovpa** command (with or without switches) by itself and without also using **ovstart**, you must first remove the OVPA local registration file (ovpa.lrf) from the list of services that are started by OV. For more information, refer to *Integrating with HP OpenView Network Node Manager*.

OPTIONS

NOTE
All of these command line switches are optional.

`-logLevel Error | Warning | Debug | Trace`

Set logging level, default value is Warning.

`-traceOutputFile FileName`

Set the output of logging messages to be a file with *FileName*.

`-traceOutputStdout`

Set the output of logging messages to be standard output. This option can be used with -traceOutputFile *filename* so that logging messages can be written both to the screen and the output file.

`-traceResync`

Enable logging messages for resync (node source) and resync parents. Logging messages contain node information including parent information OVPA is sending to Nerve Center.

To use this option, the log level must also be set to either debug or trace.

`-traceInform`

Enable logging messages for status informing from NerveCenter to OpenView. It shows status change messages from NerveCenter, whether the status change information is sent to OpenView or is logged into OpenView database.

To use this option, the log level must also be set to either debug or trace.

`-traceLoadDb`

Enable logging messages for loading nodes and interfaces from the OpenView database at initialization stage. It shows each node and each interfaces are loaded.

To use this option, the log level must also be set to either debug or trace.

`-traceParentComp`

Enable logging messages for the details of how the parents are computed for each Nerve Center host.

To use this option, the log level must also be set to either debug or trace.

`-traceMapUpdates`

Enable logging messages for the processing of OpenView events regarding changes. This option displays the significant data that arrived with an OpenView event, how ovpa is processing that event, and whether network topology needs to be updated or not.

To use this option, the log level must be set to either debug or trace.

`-traceConnects`

Enable logging messages for connections from the NerveCenter server, and from OpenView. It also shows why the port is disconnected.

To use this option, the log level must also be set to either debug or trace.

`-traceAll`

Enable logging messages for resync, inform, loadDb, parentComp, mapUpdates, and connects.

To use this option, the log level must also be set to either debug or trace.

`-pc`

Enable parent computing. Without this option, no parent information is computed for any NerveCenter host, including the requests of resync parents from NerveCenter server, and the command line option -writeParentsToFile

`-writeParentsToFile NcHostName`

When -pc is turned on, OVPA computes parent-child relationships, writes to the file NcHostName_PC.dat, and exits.

`-heartbeat seconds`

The ovpa process and the NerveCenter server process exchange heartbeats. You can configure how often the heartbeat is sent from OVPA using the -heartbeat switch. If OVPA doesn't receive a heartbeat from NerveCenter in the specified time then it will close the connection to that server's port. Default heartbeat rate is 300 seconds.

`-ignoreUnmanagedIntf`

If this switch is used OVPA does not send NerveCenter server any unmanaged interfaces within a node. You may want to restrict the interfaces that NerveCenter server must poll to only interfaces that OpenView is currently managing. If you have many subobject scope alarms and nodes, this switch to improves polling performance.

`-initNcStatusInOV`

Synchronizes the status of a NerveCenter icon and its managed nodes.

NOTE

NerveCenter must be configured so that its node data source and inform host are the same. All IP filters must be empty.

`-ignoreCapability`

Determines whether OVPA keeps track of OpenView’s capability information, such as IsRouter. For more information, see *Integrating NerveCenter with a Network Management Platform*.

-ignoreCapability prevent ovpa from keeping track of capability data. This increases ovpa’s speed and decreases its memory usage.

This switch can be turn on when a user wants to force the node to be added to NC server, regardless of capabilities.

NOTE

The -ignoreCapability switch disables NerveCenter’s ability to filter nodes based on capabilities.

`-noResolveCommunityOnResync`

When this option is turned on, OVPA does not obtain the community strings during a resync or resync parents. You can use this argument when you are sure that the community has not changed since the initial OpenView database import to improve resync performance.

`-defGlobalCommunity filename`

Instead of retrieving community strings through OpenView, you can retrieve them from a file. This option is used to give the filename where user defines default community strings.

`-preferredAddrOnly`

(OpenView only) When you start OVPA with this argument, only the preferred SNMP address is sent from OVPA to NerveCenter on resync.

If you use this argument and the preferred SNMP address for a node is changed manually from OpenView, you must restart OVPA for OVPA to function correctly.

If you use the -preferredAddrOnly option, you should start HP Openview netmon with the following command:

```
netmon -k pickSnmpAddrPolls=false
```

These netmon arguments prevent netmon from attempting to test other SNMP addresses.

`-info`

Display the version of OVPA

`-help`

Help information for OVPA

NOTES

If you want these settings to take effect every time OVPA is started, edit the file ovpa.lrf to include these switches. On Windows, the file is located in the OpenView/lrf directory. On UNIX, the file is located in the opt/OSInc/nc/OV/lrf directory. For example, to have OVPA compute parenting information only during a resync you would modify ovpa.lrf to read:

```
ovpa:/opt/OSInc/nc/OV/bin/ovpa:
OVs_YES_START:ovwdb:-pc_noupdate:OVs_WELL_BEHAVED:5:
```

paserver

NAME

paserver

SYNOPSIS

```
paserver [-d] [-g] [-h|-?] [-n ON|OFF] [-nhost] [-nport] [-o] [-p]
            [-r] [-scm {a|m|r|s}] [-t] [-tcfg] [-thost] [-toneway]
            [-tport] [-ttype] [-u ON|OFF] [-v]
```

DESCRIPTION

Runs the NerveCenter Universal Platform Adapter for integrating NerveCenter with IBM Tivoli Netcool/OMNIbus.

OPTIONS

`-d`

Runs the Universal Platform Adapter from the command line in debug mode and outputs debug messages to the console. The next time the host machine boots, the Universal Platform Adapter will run as a service or daemon again.

`-g`

(Windows only) Registers the Universal Platform Adapter as an Event Source.

`-h | -?`

Displays command line help for the Universal Platform Adapter switches.

`-n`

Enables or disables NerveCenter integration with IBM Tivoli Netcool/OMNIbus.

On Windows, when starting paserver from the command line, you must specify either **-d** or **-scm** *option* in combination with either **-n ON** or **-u ON**.

`-nhost`

Defines the machine on which the Netcool probe is located. The default is localhost.

`-nport`

Defines the port the NerveCenter platform adapter uses to communicate with IBM Tivoli Netcool/OMNIbus. The default is 32510.

`-o`

(Windows only) Records values into Registry. Any options (other than -scm) become a part of the standard configuration.

To use this switch, you should first stop the Universal Platform Adapter. You must restart the Universal Platform Adapter.

`-p`

Defines the platform adapter's listening port. The default is 32509.

NOTE
This number must match the one in NerveCenter Administrator.

`-r`

(Windows only) Shortened version of -scm r. Removes the Universal Platform Adapter as a service. It also removes Registry entries created at install time.

`-scm a|m|r|s`

(Windows only) Changes settings in the service control manager.

`a`

(Windows only) Installs the Universal Platform Adapter as a service, marking it as autostart. The service will start following this command.

`m`

(Windows only) Installs the Universal Platform Adapter as a service or daemon, marking it as start on demand. The service will not start following this command.

`r`

(Windows only) Removes the Universal Platform Adapter as a service. If it the service is running, it will be stopped.

`s`

(Windows only) Starts the Universal Platform Adapter as a service. This may be combined with **a** or **m**.

`-tcfg`

The full qualified path/filename for the Event Adapter configuration file. The default is /opt/OSInc/userfiles/nctec.cfg

`-thost`

The machine on which the Tivoli Event Server is located. The default is localhost.

`-tport`

The port the NerveCenter platform adapter uses to communicate with Tivoli Event Server. The default is 32510.

`-ttype`

The type of connection to be established with the Tivoli Event Server:

- 0 Connectionless delivery to agent.
- 1 Connection-oriented (secure) delivery to agent.
- 2 Default type of connection.

The default is 0.

`-v`

Views current Universal Platform Adapter settings.

NOTES

If you want these settings to take effect every time the Universal Platform Adapter is started, edit the file pastart to include these switches. pastart resides in NerveCenter \bin directory.

On Windows, when starting paserver from the command line, you must specify either **-d** or **-scm** *option* in combination with either **-n ON** or **-u ON**.

serializedb

NAME

serializedb

SYNOPSIS

```
serializedb
-f fullyQualifiedFilename.asc
-o {dbtofile|filetodb}
{-r}|{-c “DSN=dataSourceName;UID=userID;PWD=password”}|{-d
fullyQualifiedFilename.ncdb}
```

DESCRIPTION

Exports or imports the data in your NerveCenter database. You can run SerializeDB from a GUI, the command line, or from a script.

CAUTION

Before you import from or export to a database, you must shut down the NerveCenter Server. If you don’t shut down the NerveCenter Server, you could lose data. If you are importing from or exporting to a UNIX flat file database, all of the files that make up the database (NerveCenter.ncdb and NerveCenter.node) must be in the same directory.

options

`-f` *`fullyQualifiedFilename`*`.asc`

Name of the backup file you want SerializeDB to export to or import from. If you do not specify a path, SerializeDB creates (or searches) for the file in your current working directory.

`-o dbtofile or filetodb`

Specifies whether you are using SerializeDB to export or to import NerveCenter data.

`-r`

(Windows only) Instructs SerializeDB to get the data source name, user ID, and password from the NerveCenter registry. You can use -r with a Microsoft SQL Server database or a Microsoft Access database.

`-c “DSN=dataSourceName;UID=userID;PWD=`*`password`*`”`

(Windows only) Specifies that you are exporting from a Microsoft SQL Server database. Replace *dataSourceName* with the existing data source name, *userID* with the appropriate SQL Server user ID, and *password* with the corresponding password.

`-d` *`fullyQualifiedFilename`*`.ncdb`

(UNIX only) Specifies the fully qualified database pathname (for example, /opt/OSInc/db/NerveCenter.ncdb). If you don’t specify a path, NerveCenter uses the current working directory.

USAGE

⚠ **CAUTION**

On UNIX, SerializeDB requires an Xdisplay even when running in command line mode. Therefore, do not run SerializeDB remotely or on a headless server via TTY.

Unless you use fully qualified pathnames, SerializeDB reads and writes to the current working directory. When running SerializeDB from the command line, you can only export to a serialized file (.asc file).

Use the timestamp and the size of the file as an indicator of success. For example, if the timestamp is old or the file size is 0K, you need to investigate further. Check the UNIX system log or the Windows Event Log for error messages.

trapgen

NAME

trapgen

SYNOPSIS

```
trapgen {-v1 | -v2c | -v2cinform | -v3inform} [-p port] [-c community]
           [-w [-t timeout] [-r retries] ] destinationAddress
           enterprise {agentAddress | trapOID}
           {genericTrapNumber | specificTrapNumber} timeStamp
           NoAuth|MD5|SHA NoPriv|DES username [variable type value...]
```

DESCRIPTION

TrapGen is a standalone utility that allows you to send an SNMP trap or inform to a particular device. From a command line or shell prompt, you can set the generic and specific trap numbers, provide the sender and recipient IP addresses, include an enterprise identifier, and specify the variable bindings.

Generating a trap is useful for testing trap masks and behavior models. When you send an SNMP trap to a NerveCenter Server, any trap mask configured to detect the specified trap values fires its trigger. If your trap mask and alarm are working properly, you see the alarm listed in your console.

OPTIONS

`-v1`

Sends a SNMPv1 trap. This is the default

`-v2c`

Sends a SNMPv2c trap.

`-v2cinform`

Sends a SNMPv2c inform.

`-v3inform`

Sends a SNMPv3 inform.

`-p` *`port`*

Port number to which the trap is being sent. The default is 162.

`-c` *`community`*

The community name for the agent sending the trap. The default is "public".

`-w`

Use with -v2cinform and -v3inform to wait for a response back from the machine to which the inform is being sent.

`-t timeout`

The time in hundredths of a second for which Trapgen waits for a response from an inform. The default is 300 (3 seconds).

`-r retries`

The number of times Trapgen will attempt to re-send an inform if the inform does not respond. The default is 0.

`destinationAddress`

Destination address for the trap. *destinationAddress* can be an IP address or a node’s name.

`enterprise`

(v1 only) SNMP enterprise-specific trap number in dot notation (such as 1.3.6.1.4.1.9) that identifies the agent sending the trap. You can specify a null value (“”) for *enterprise*. The null value equates to 1.3.6.1.4.1.78.

`agentAddress`

(v1 only) Agent sending the trap. *destinationAddress* can be an IP address or a node’s name. You can specify a null value (“”) for *agentAddress*.

`trapOID`

(v2c, v2cinform, and v3inform only) A unique object identifier for the trap itself.

`genericTrapNumber`

(v1 only) One of the SNMP generic trap numbers (0-6).

- 0—coldStart
- 1—warmStart
- 2—linkDown
- 3—linkUp
- 4—authenticationFailure
- 5—egpNeighborLoss
- 6—enterpriseSpecific

`specificTrapNumber`

(v1 only) A particular enterprise-specific trap number. If not used, specify zero.

`timeStamp`

Time elapsed between the last (re)initialization of the entity and the generation of the trap in hundredths of seconds. Taken from the MIB-II object, **system.sysUpTime**. You can specify a null value (“”) for *timeStamp*. The null value equates to 1.

`NoAuth | MD5 | SHA`

(v3inform only) The authentication protocol. If you are sending a v3inform, you must supply an authentication protocol.

`NoPriv | DES`

(v3inform only) The privacy protocol, required if you are sending a v3inform.

`username`

(v3inform only) Name of a user configured on the remote agent. The default is ncuser.

`variable`

Name of *variable*. *variable* must be a valid OID in dot notation. A string is not acceptable. For example, use 1.3.6.1.2.1.1.1.0 instead of sysDescr.0.

`type`

Type for *variable*. *type* can be one of: integer, integer32, uinteger32, octetstring, objectidentifier, ipaddress, counter, counter32, counter64, gauge, gauge32, or timeticks.

`value`

Value of *variable*. Enclosing strings with quotes is optional.

EXAMPLES

Example 1: In the following example, Trapgen sends an generic SNMPv1 authentication failure trap from the NerveCenter Server Blueridge to itself:

```
trapgen blueridge “” blueridge 4 0 “”
```

enterprise and *timeStamp* are nulls (“”) and *specificTrapNumber* is 0 since a trapgen is sending a generic trap.

Example2: In the following example, Trapgen sends to Blueridge a v2c trap identified with a unique OID (1.3.6.1.2.1.1.1), from an agent (1.3.6.1.4.1.9.1.3) using a null timestamp:

```
trapgen -v2c blueridge 1.3.6.1.4.1.9.b1.3 1.3.6.1.2.1.1.1 “”
```

Example 3: In this example, Trapgen sends to the node 10.52.174.139 a specific trap number (110000) from the node poe. The trap contains two varbinds with a values of type timeticks and octetstring, respectively.

```
trapgen 10.52.174.139 1.3.6.1.4.1.9.1.3 poe 6 110000 900
1.3.6.1.2.1.1.3 timeticks 9875632 1.3.6.1.2.1.1.1 octetstring “Cisco
trouter”
```

Example 4: In this example, Trapgen sends an SNMPv3 Inform to a remote NCServer at 10.69.89.51. The remote NCServer is setup with the default SNMPv3 user (ncuser). A null time stamp is used.

```
trapgen -v3inform -w 10.69.89.51 1.3.6.1.4.1.78 1.3.6.1.4.1.78.3.5 “”
NoAuth NoPriv ncuser
```

Traprcv

NAME

trapgen

DESCRIPTION

The traprcv command displays the SNMP Trap messages received by the NerveCenter Trap service. This utility can be useful when debugging behavior models. When you run the traprcv command from a command line or shell prompt, any trap you receive is shown on the screen.

Traprcv can receive the following traps and informs:

- SNMPv1 Traps
- SNMPv2c Traps
- SNMPv2c Informs
- SNMPv3 Traps
- SNMPv3 Informs

Command Line Interface Reference

You can use NerveCenter commands to control NerveCenter servers, nodes, alarms, polls, and masks from the command line interface from a scriptfile, single command, or interactive session.

The commands are generally made up of an action and an object on which you perform the action. For example, the command **list mask** is made up of the list action, which is performed on the mask object. Command line switches are case-insensitive.

This reference consists of the following sections:

- *Server Commands on page 319*
- *Object Commands on page 322*
- *General Commands on page 329*
- *Notes for CLI Commands on page 330*
- *CLI Command Examples on page 330*

Server Commands

Use these commands to list, select, disconnect from, or display information about connected NerveCenter Servers in your CLI session. These commands are listed in the order that they appear in the syntax diagram.

```
serverCommand [-serverSwitch [parameter]] [-serverSwitch [parameter]]...|-?
```

serverCommand

One of the several server-related keywords and their switch/parameter pairs described below. *serverCommand* also includes the connect command and its switch/parameter pairs described in *OPTIONS* of *nccmd on page 300*.

NerveCenter directs server commands to the active server if no server is specified. The active server is the last server connected to or selected using the select server command.

TABLE C-2. Server Commands

serverCommand	Description	Options

`clear trigger -t` *`triggerName`* `[-i` *`alarmInstanceID`*`]`	Cancels any pending *triggername* triggers that were queued by its own alarm instance. If no alarm instance is supplied, then *triggername* is canceled for all alarms in which it is pending.	`-t` *`triggerName`*
		Name of the trigger to be cleared.
		`-i` *`alarmInstanceID`*
		(Optional) *alarmInstanceID* is a numeric value indicating the alarm instance for which NerveCenter is clearing the specified trigger.

`disconnect`	Disconnects from the active server.	

`fire trigger -t` *`triggerName`* `-n` *`nodeName`*`|$ANY -o` *`subobjectName`*`\|$ANY [-d` *`delay`*`]`	Fires the specified trigger. If no delay is supplied, then NerveCenter fires the trigger immediately.	`-t` *`triggerName`*
		Name of the trigger to be cleared.
		`-n` *`nodeName`*`|$ANY`
		Name of the node on which to fire the trigger or any node ($ANY).
		`-o` *`subobjectName`*`|$ANY`
		The name of the subobject on which the trigger is fired or any subobject ($ANY).
		`-d` *`delay`*
		(Optional) *delay* is the number of seconds before NerveCenter fires the specified trigger.

`list`	Lists the connected servers.	
`select -s` *`serverName`*	Makes a connected *serverName* the active server.	

`set server {-t` *`timeoutInterval`*`}|{-w` *`width`*`}|-?`	Allows you to configure how the server commands respond.	`-t` *`timeoutInterval`*
		timeoutInterval is the number of seconds that the CLI waits for the NerveCenter server to acknowledge a CLI request. The default is five seconds (UNIX) and one second (Windows). -t *timeoutInterval* is commonly used in situations where a server is busy or the connection is slow. Should a timeout occur, the connection is still maintained, but the CLI displays the message: **Error. No data was received within allowed time**.
		`-w` *`width`*
		width is the column width (1-50 characters; 12 is the default) for tabular output from the **list** command.
		`-?`
		Displays help for the **set server** commands.

-switch parameter

One of several switch/parameter pairs described below. Parameter switches are case-insensitive and may be preceded by either a hyphen (-) or a backslash (\).

```
show server -serverSwitch
```

Displays information about the active server.

TABLE C-3. Server Switches

-serverSwitch	**Description:**
-a	Display information about the NerveCenter administrators connected to the active server.
-c	Display information about the NerveCenter clients connected to the active server.
-d	Display information about the NerveCenter database.
-h	Display information about the Inform host.
-i	Display information about the active server.
-l	Display information about the NerveCenter license key.
-m	Display information about the map host.
-s	Make *serverName* the active server. Used only with the *serverCommand* **select**.
-?	Display help information about the specified *serverCommand*.

Object Commands

Use these commands to delete, list, or set alarms, trap masks, nodes, and polls. These commands are listed in the order that they appear in the syntax diagram.

```
{add | delete | list object {-switch parameter} | -? } |
{set object {-switch parameter -w targetObject} | -? }
```

add node

Adds the specified node. A unique node name (`-n`) and at least one IP Address (`-i`) must be specified. All other parameters are optional and will be set to a default value if omitted.

Usage

```
add node -n (NodeName) -i (IpAddress) [-i IpAddressN] [-p PortNumber]
    [-m (yes|no)] [-s (yes|no)] [-d (yes|no)] [-g PropertyGroup]
    [-v (1|2|3|u)] [-c ReadCommunity] [-t WriteCommunity]
    [-3 (User1|User2|Local)] [-u v3username] [-z v3context]
    [-l (NoAuthNoPriv|AuthNoPriv|AuthPriv)]
    [-a (SHA|MD5) ] [-h (AuthPassword)]
    [-o (DES|3DES|AES128|AES192|AES256)] [-y (PrivacyPassword)]
```

`add -?`

Returns possible options for the 'add' action command.

`add node -?`

Returns the syntax description of the add node command.

Required Switches

`-n NodeName`

A unique node name for the server. Node names are not case-sensitive.

`-i IpAddress`

The IP Address of the node, specified in dotted format. To specify multiple addresses, use multiple *-i* flags with a single address following each. At least one address must be specified. The validity or uniqueness of the IP address is not verified.

Optional Switches

`[-p SnmpPortNumber]`

Set the port number for SNMP operations of the new node to the specified value. If this switch is omitted the node will be set to port 161.

`[-m (yes|no)]`

Set the new node to be managed (yes), or not managed (no). The default is yes.

`[-s (yes|no)]`

Set the new node to be suppressed (yes), or not suppressed (no). The default is no.

`[-d (yes|no)]`

Set the new node to be eligible for automatic delete (yes), or not eligible (no). The default is yes.

`[-g PropertyGroupName]`

Set the property group of the new node to the specified value. The PropertyGroup must already exist for the server. If this switch is omitted the PropertyGroup of the node will be set to “NCDefaultGroup”. PropertyGroup names are case-sensitive.

`[-v (1|2|3|u)]`

Set the SNMP version of the new node to the specified value. The default is version 1.

Optional Switches, SNMPv1 or v2 (`[-v 1]` or `[-v 2]`)

`[-c ReadCommunity]`

Set the read community string of the new node to the specified value. This switch will be ignored unless the SNMP version (-v flag) is specified as '1', '2', or 'u'. If this switch is omitted and the node is not version 3 the node will be set to read community 'public'.

`[-t WriteCommunity]`

Set the write community string of the new node to the specified value. This switch will be ignored unless the SNMP version (-v flag) is specified as '1', '2', or 'u'. If this switch is omitted and the node is not version 3 the node will be set to write community 'public'.

Optional Switches, SNMPv3 (`[-v 3]`)

`[-3 (User1|User2|Local)]`

Sets the SNMPv3 type. The default is Local.

`[-l (NoAuthNoPriv|AuthNoPriv|AuthPriv)]`

Set the security level of the new node to the specified value. This switch will be ignored unless the SNMP version (-v flag) is specified as '3'. If this switch is omitted for version 3 nodes, the security level defaults to 'NoAuthNoPriv'.

Optional Switches, SNMPv3 Local definition (`[-v 3 -3 Local]`)

`[-u v3username]`

Sets an SNMPv3 user name.

`[-z v3context]`

Sets the SNMPv3 context. The default is `-z ""` (empty).

`[-a (SHA|MD5)]`

Sets the authentication protocol of the new node to the specified value. This switch will be ignored unless the SNMP version (-v flag) is specified as '3' and the security level has been specified as 'AuthPriv' or 'AuthNoPriv'. If this switch is omitted for version 3 nodes, the authentication protocol defaults to SHA.

`[-h (AuthPassword)]`

Sets an SNMPv3 passkey.

`[-o (DES|3DES|AES128|AES192|AES256)]`

Sets the SNMPv3 privacy mode. The default is DES.

`[-y (PrivacyPassword)]`

Sets the SNMPv3 privacy passkey.

The add node command indicates whether or not the node was added. If successful, the text "'NodeName' was added" is returned. If a failure occurred, the text "Error" is returned, followed by the reason for the failure.

Rules

1. The node name must be provided. `[ -n (NodeName) ]`
2. The node name must not already be in the server's node list.
3. At least one IP Address `[ -i IpAddress ]` must be given.
4. The Property Group, if specified, must be known to the server

Examples

The following five examples demonstrate the basic command structure.

- The simplest form of "add node":

```
NC> add node -n gateway -i 192.168.1.1
```

- The same command, with argument aliases (see "`add node -? aliases`"):

```
NC> add node -nodename gateway -ipaddr 192.168.1.1 -ipaddr
    194.22.19.251
```

- Adding a SNMPv2c router where the Property Group is Mib-II-switch:

```
NC> add node -n switch -I 192.168.1.100 -g Mib-II-switch -v 2
    -c public -t dontTouchThis
```

- Adding a SNMPv3 device, providing the v3 communication configuration:

```
NC> add node -n archiver -i 192.168.1.23 -g Mib-II -v 3 -c public
    -3 Local -u Admin -l authNoPriv -a SHA -h lArg3rThAnL1f3
```

- Adding a node that has two IP addresses but is not to be monitored:

```
NC> add node -n ncserver -i 127.0.0.1 -i 192.168.1.30 -g Mib-II
    -m no -d no
```

The next four examples demonstrate how to create different SNMP node types, with the command output included after each step:

- Adding an SNMPv1 node:

```
NC> add node -n Zeus -i 10.150.1.2 -m yes -s no -d no -g Mib-II
    -v 1 -c public -t private

Zeus was added

NC> list node -n Zeus

Name       Group      Severity   Managed    Suppressed  SNMPVer.
Zeus      Mib-II     Normal     Yes        No          V1
No. of Object(s) = 1
```

- Adding an SNMPv2 node:

```
NC> add node -n Athena -i 10.150.1.3 -m yes -s no -d yes -g Mib-II
   -v 2 -c public -t owl

Athena was added

NC> list node -n Athena

Name     Group    Severity   Managed    Suppressed  SNMPVer.
Athena   Mib-II   Normal     Yes        No          V2
No. of Object(s) = 1
```

- Adding a node where the SNMP version is specified as Unknown:

```
NC> add node -n Aries -i 10.150.1.4 -m no -d yes -s yes -g Mib-II -v u

Aries was added
```

```
NC> list node -n Aries

Name       Group      Severity     Managed      Suppressed  SNMPVer.
Aries      Mib-II     Normal       No           Yes         Unk.
No. of Object(s) = 1

NC> list -node -g Mib-II

Name         Group        Severity     Managed      Suppressed  SNMPVer.
Aries       Mib-II      Normal      No          Yes         Unk.
Athena      Mib-II      Normal      Yes         No          V2
Zeus        Mib-II      Normal      Yes         No          V1
No. of Object(s) = 3
```

- Adding an SNMPv3 node with "MotherEarth" as the user name, with the context "EarthMother", and the NoAuth/NoPriv Security Level:

```
NC> add node -n Hera -i 10.150.1.5 -m yes -v 3 -3 Local
    -u MotherEarch -z EarthMother -l NoAuthNoPriv

Hera was added

NC> list node -g Mib-II

Name       Group        Severity     Managed      Suppressed  SNMPVer.
Aries      Mib-II      Normal      No          Yes         Unk.
Athena     Mib-II      Normal      Yes         No          V2
Zeus       Mib-II      Normal      Yes         No          V1
No. of Object(s) = 3

NC>list node -n *

Name         Group        Severity     Managed      Suppressed  SNMPVer.
Hera        NCDefault.. Normal      Yes         No          V3
Aries       Mib-II      Normal      No          Yes         Unk.
Athena      Mib-II      Normal      Yes         No          V2
Zeus        Mib-II      Normal      Yes         No          V1
No. of Object(s) = 5
```

delete

Deletes the specified *object*. One switch/parameter pair is required.

`list`

Lists the specified alarm, mask, node, or poll when specified with a switch/parameter pair. When used without a switch/parameter pair, lists the connected and active servers (if any).

`object`

Specifies the type of NerveCenter object that you are deleting, listing, or setting attributes for. *object* can be: alarm, mask, node, or poll.

`-switch parameter`

One of several switch/parameter pairs described below. Parameter switches can be preceded by either a hyphen (-) or a backslash (\).

`-a name | *`

Identifies a specific NerveCenter alarm or poll (*name*) or all (*) alarms or polls. *name* is case-sensitive. -a does not apply to trap masks. See **-m**, **-n**, and **-p**.

`-a authenticationProtocol`

Identifies NerveCenter nodes by their authentication protocols. Valid *authenticationProtocol* values are: MD5 (Message Digest algorithm) and SHA (Secure Hash Algorithm). The default protocol is MD5. For privacy, the CBC-DES protocol will be used, since this is the default provided by Envoy.

`-c communitystring`

Identifies the SNMPv1/v2c Read Community String for nodes. Used by the 'set node' and 'add node' commands.”

`-e on | off`

Identifies NerveCenter objects by their state: enabled (on) or disabled (off).

`-g propertyGroup`

Identifies NerveCenter nodes by their property group.

`-l securityLevel`

Identifies NerveCenter nodes by their SNMPv3 security level. Valid *securityLevel* values are: NoAuthNoPriv (no authentication protocol, no privacy password), AuthNoPriv (authentication protocol, no privacy password), and AuthPriv (authentication protocol, privacy password).

`-m name | *`

Identifies a specific NerveCenter trap mask (*name*) or all (*) trap masks. *name* is case-sensitive. -m only applies to trap masks. See **-a**, **-n**, and **-p**.

`-m yes | no`

Specifies NerveCenter managed nodes (yes) or unmanaged nodes (no).

`-n` *`name`* `| *`

Identifies a specific NerveCenter node (*name*) or all (*) nodes. *name* is case-sensitive. To display extended SNMPv3 attributes (security level, authentication protocol, and error status), use **-x v3**. See **-a**, **-m**, and **-p**.

`-p` *`name`* `| *`

Identifies a specific NerveCenter poll (*name*) or all (*) polls. *name* is case-sensitive. See **-a**, **-m**, and **-n**.

`-r` *`propertyName`*

Identifies NerveCenter alarms or polls by their property. *propertyName* is case-sensitive.

`-r` *`errorStatus`*

Identifies NerveCenter nodes by their error statuses. Valid *errorStatus* values are: AuthKeyFail, AuthPrivKeyFail, V3InitFail, ClassifyFail, TimeSyncFail

`-s enterprise | node | subobject | instance`

Identifies NerveCenter alarms by their scope.

`-s yes | no`

Identifies NerveCenter nodes their state: suppressed (yes) or unsuppressed (no).

`-t communitystring`

Identifies the SNMPv1/v2c Write Community String for nodes. Used by the 'set node' and 'add node' commands.”

`-v` *`SNMPVersion`*

Identifies NerveCenter masks, nodes, or polls by the version of the SNMP agent. Valid *SNMPVersion* values are: 1, 2, or 3.

`-w` *`targetObject`*

Flag indicating that target object follows for the set command.

The arguments preceding the -w switch determine which attributes of an object(s) are set and what they are set to. Those that follow **-w** determine which object(s) are modified. For example, the sequence **-r** *property1* **-w -r** *property2* would change the property of all objects with property2 to property1.

`targetObject`

Switch/parameter pair that identifies NerveCenter objects whose attributes are being set.

`-x v3`

Displays all SNMPv3 nodes' extended attributes (security level, authentication protocol, and error status), which are not displayed by the **list node -n** command.

`-?`

Display help information about the delete/list/set *object* command.

`set`

Sets attributes for a specified object. One switch/parameter pair and **-w** *targetObject* is required.

General Commands

Use these commands to get command line help, quit the NerveCenter CLI, or to resync with the network management platform. These commands are listed in the order that they appear in the syntax diagram.

```
-? | quit | reset alarm_instance -i instanceNumber [-s stateName] | resync
```

`-?`

Displays help or, if used without an action keyword or *object*, displays general information about help. If used with an action alone, gives general command information such as syntax and the objects to which it can be applied. If used with an action and *object*, gives detailed information about the command's syntax.

`quit`

Ends the CLI session and automatically disconnects from each NerveCenter Server.

`reset alarm_instance`

Transitions the specified alarm to the ground state.

`-i instanceNumber [-s stateName]`

Identifies the alarm to be rest to ground by a valid alarm instance number (required) and the state of the alarm (optional).

`resync`

Updates the NerveCenter node list with information from the network management platform. NerveCenter automatically executes a resync at startup and when you reconnect to the platform after the connection was broken.

Notes for CLI Commands

If you leave out any mandatory parameters, nccmd returns an error. In general, nccmd also returns an error if it encounters any syntax errors (for example, unknown or incorrectly specified parameters) and does not issue the command to the connected server. Specific error messages are returned by nccmd if the server encounters errors.

CLI Command Examples

The following command starts a CLI interactive session, connecting to the Blueridge server, with the user ID GRakauskas, and using port 32506:

```
nccmd connect -s blueridge -u GRakauskas -n 32506
```

This command sets the property to mynodes for all alarms that are enabled:

```
set alarm -r mynodes -w -e on
```

The command that follows assigns all nodes belonging to the troublemaker property group to the criticaldevices property group:

```
set node -g criticaldevices -w -g troublemaker
```

In this example, all alarms set to enterprise scope are displayed:

```
list alarm -s enterprise
```

Glossary

D

Action Router	Performs standard NerveCenter alarm actions in response to detected network conditions and based on additional conditions that you specify. In NerveCenter Client, you first define a set of conditions along with the actions to be completed. Then, you can assign the Action Router action to an alarm transition. When the transition occurs, NerveCenter evaluates all existing conditions to determine whether any have been met and, therefore, whether one or more conditional actions should be performed. All alarms that are sent to the Action Router are evaluated against all the current rules. See also *alarm action*.
Action Router rule	An Action Router rule is composed of a conditional statement and one or more actions. The conditional statement can contain one or more conditions logically joined together.
Administrator rights	Users with administrator logon rights can customize NerveCenter and save changes to the NerveCenter database. In the NerveCenter Client, administrator rights are required to create or modify the objects used in behavior models. Those with administrator rights belong to the NerveCenter Admins group on Windows NT or to ncadmins on UNIX. The can also log in as administrator on Windows NT or root on UNIX. See also *user rights*.
alarm	A NerveCenter object that detects a trigger generated by a poll, trap mask, Perl subroutine, or other alarm. An alarm is a finite state machine that transitions from one state to the next and performs any actions assigned to the transition. Each transition is triggered by its own set of network data, which is defined in the associated trigger generator. When an alarm detects its first trigger, the alarm transitions to the next state, where it remains until another trigger is received—either from the same or another trigger generator. The sequence of transitions enables NerveCenter to monitor persistent, simultaneous, or sequential events that, taken together, could indicate a critical or important condition.

D Glossary

alarm action	A NerveCenter automated response that helps you manage network activity and stay informed about network conditions. You can assign one or more actions to any transition in an alarm state diagram. Actions fall into four main categories: notification, logging, triggering other alarms, and correcting network conditions. Examples include sending e-mail, issuing a page, logging data, sending an SNMP trap, executing a command, and sending an Inform message to a network management platform. In addition, NerveCenter actions can be performed conditionally based on criteria that you specify using the Action Router.
alarm circular transition	A circular transition is one in which the From State and To State are the same for a transition in an alarm state diagram. When a trigger containing the polled data arrives at this transition, the transition occurs and returns the alarm to the same state. The effect is that the alarm continues looping with each instance of the trigger. This type of transition usually includes an alarm counter action that tracks the number of occurrences and fires a trigger when the number reaches a certain value.
alarm instance	A single instance of a detected network event. Each instance is one active occurrence of an alarm definition that tracks a current network or system condition through its own copy of the alarm's state diagram. For example, one alarm might have five distinct alarm instances, each tracking the same condition on a different node. See also *alarm scope*.
alarm instance ID	The unique identifier for an alarm instance managed by a NerveCenter Server. The instance ID is listed in the Alarm History window for an alarm instance. NerveCenter includes a variable $AlarmInstanceID that you can use in Perl subroutines, command actions, and log actions to specify the instance ID associated with an alarm.
alarm scope	A setting that determines whether an alarm instance monitors a subobject (for example, a port), several MIB objects on a subobject, a node, or an entire enterprise. Scope is assigned in an alarm's definition window. If an alarm is using node scope, each alarm instance tracks the alarm's states for a single, distinct node. If an alarm is using subobject scope, each instance tracks the alarm's states for a MIB base object on a node, for example, an interface on a router. Instance scope alarms track instances for every interface or port that fits the polled condition regardless of the base object. Enterprise scope alarms track an event for the enterprise as a whole.

alarm severity	An indication of the urgency of a detected event. When creating an alarm, you assign each state a severity level, which is defined by a name and unique color. NerveCenter ships with severity options ranging from normal to critical for a fault condition, and from very low to saturated for a traffic condition. If your network management platform has event severities, you can map NerveCenter's severities to match those on your platform.
alarm state	Corresponds to a network condition that an alarm is monitoring. To monitor certain network conditions, or states, you must specify these states in the alarm's state diagram. Each state listens for certain triggers. Once the correct trigger is fired, the alarm transitions to the corresponding state.
alarm state diagram	Specifies which detected conditions are correlated, in what order, and which responses (if any) are assigned to each stage. For each alarm that you create, you design a state diagram to specify the states you want to monitor. The state diagram can detect persistent, simultaneous, or sequential events that, taken together, indicate a critical or important network condition. Incoming triggers transition the alarm from one state to another.
alarm summary views	NerveCenter Client provides two views for monitoring network activity. Both windows can be opened from the Admin menu or the toolbar: **Alarm Summary** - In the Alarm Summary window, you monitor alarm instances for the connected server. If you are connected to more than one server, you can change the active server and view the Alarm Summary window for the new server. To do this, select the server you want to view from the Active Server drop-down listbox on the NerveCenter toolbar. **Aggregate Alarm Summary** - In the Aggregate Alarm Summary window, you can monitor alarm instances collectively for all servers to which you are connected.
alarm transition	A change from one alarm state to another, prompted by a *trigger*. When an alarm transitions from its ground state, the transition creates an alarm instance. Further triggers affect the current alarm instance, each generating a transition in the alarm. During a transition, NerveCenter carries out any alarm actions that are assigned to the transition.
ASN.1 file	MIB base objects are defined using Abstract Syntax Notation One (ASN.1), a language that's understood by network management protocols. The ASN.1 language is described in the ISO documents ISO.8824 and ISO.8825. See also *Management information base (MIB)*.

D Glossary

attribute	NerveCenter terminology includes two meanings for the word attribute: A MIB object that contains an actual value. For example, sysContact, ifInOctets, and ipInDiscards are all attributes in the industry-standard MIB-II because they contain strings, numbers, or other values. The value associated with an attribute can be static (for example, the speed of the interface) or dynamic (for example, an entry in a routing table). A value that you assign to a NerveCenter object. For example, if you assign a property to a poll, that property is an attribute of the poll. NerveCenter's Set Attribute alarm action allows you to change certain node, poll, trap mask, or alarm attributes when a specified network condition is detected.
authentication	In SNMPv3 communication, the process of confirming the parties (i.e. entities) that communicate with each other as well as the timeliness of the messages received at an SNMPv3 entity.
authentication key	The seed used to generate a message digest, as specified in the authentication protocols, to authenticate SNMPv3 messages. Both sender and receiver generate the digest, and the receiver matches the generated digest against the digest accompanying the message received to authenticate the message.
authentication protocol	Protocol used with SNMPv3 communication that allows you to verify the sender and timestamp of a message. Two authentication protocols are currently defined: HMAC-MD5-96, which is based on MD5, and HMAC-SHA-96, which is based on SHA-1 cryptographic algorithms. NerveCenter supports both protocols.
AuthNoPriv	Security level that requires message authentication services to be used while communicating with an SNMPv3 entity.
AuthPriv	Security level that requires both the message authentication and message encryption services to be used while communicating with an SNMPv3 entity.
base object	See *MIB base object*.
behavior model	The group of all alarm, trigger generator, and property group definitions required to detect and handle a particular network or system behavior. A typical behavior model consists of an alarm with all its supporting trigger generators, though behavior models can have multiple alarms. Any NerveCenter object can be associated with one or more behavior models. You can customize the behavior models that ship with NerveCenter as well as create new behavior models.
boots	See engine boots.

CBC-DES	A privacy protocol as specified in SNMPv3 specifications to be used for message encryption in communication between two SNMPv3 entities with AuthPriv security level. This protocol is supported by NerveCenter.
context	Every communication between two SNMPv3 entities takes place on behalf of a *user* (a uniquely identified entity in the SNMPv3 management domain), and potentially in some *context* (a uniquely identified entity for access control generally configured on conventional SNMPv3 agents/nodes) with any one of three security levels. These three parameters (user, context, and security level) are used together by VACM (View based Access Control Mechanism defined in SNMPv3 specifications) to grant access to any MIB data at the agents/nodes. Context can be thought of as the MIB information available to a particular user who seeks information from an agent using a certain security level. NerveCenter requires a context only if defined by the agent.
correlation expression	A Boolean expression you can use to create alarms.
DBWizard	The utility that enables you to create or convert a NerveCenter database on Windows. Run the Database Wizard, DBWizard.exe, from your Start menu. See also *SerializeDB*.
DES	A privacy protocol as specified in SNMPv3 to be used for message encryption in communication between two SNMPv3 entities with AuthPriv security level. This protocol is supported by NerveCenter. See also CBC-DES.
digest	The hashing code generated for message authentication using authentication key. This digest is appended to the message being sent out by a sending SNMPv3 entity. The receiving SNMPv3 entity separate out this digest from the message, generate the digest again from the message using the locally available authentication key and then compares the two digests for message authentication.
discovery	The process of discovering nodes and adding them to the NerveCenter database. When you enable discovery, if the NerveCenter database does not already contain a node matching the source of an SNMP trap or NerveCenter Inform, it adds that node to the database. You can also customize NerveCenter's Discovery behavior model, which uses a TCI/IP sweep program (ipsweep.exe) to populate your node list. Alternatively, the NerveCenter node list can be populated from your network management platform.
Downstream alarm suppression	A NerveCenter behavior model that uses parent-child topology information to determine the status of nodes and suppress polling on nodes that are down.
engine boots	Number of times an SNMPv3 engine has been started or re-initialized (i.e., booted) since its engine ID was last configured.

D Glossary

engine ID	An SNMPEngineID value with a length of 12 octets that uniquely identifies an SNMPv3 engine within the SNMPv3 management domain. There is one SNMP engine ID for every instance of a NerveCenter Server.
engine information	An SNMPv3 agent's engine ID, boots, and time ticks.
engine time ticks	As defined in SNMPv3 specifications, number of seconds elapsed since the last engine boot. When this counter reaches its maximum limit, it is reset to 0 and the engine boots value is incremented by 1.
enterprise scope	A setting for an alarm definition that determines that there will be at most one alarm instance monitoring the entire enterprise. See also *alarm scope*.

escape character	If you enter an expression that contains a period, NerveCenter interprets the string as if it were a BaseObject.Attribute. In order to use a period in any other context, you must use an escape character (\) with the period. This applies to filenames, IP addresses, or any other string containing a literal "." character. For example, if your script refers to a fully qualified host name, you might enter something like the following:
	
	if ($nodeName ne "myhost\.mydomain\.com" {
	FireTrigger (mytrigger);
	}

event	A detected network or system condition. An event can be forwarded to one or more NerveCenters and network management platforms.
generic trap number	Indicates the nature of an SNMP trap detected by a NerveCenter trap mask. There are seven different types of traps, numbered 0 through 6. The first six (0-5) are industry-standard traps and are described in any standard SNMP text. NerveCenter also provides an option for selecting all six traps. Trap number 6 is reserved for enterprise-specific traps and NerveCenter Inform actions. See also *specific trap number*.
Hewlett Packard OpenView Network Node Manager	A part of the OpenView network management platform that provides network discovery, a map, threshold detection, a MIB browser, an event browser, and other functions. NerveCenter integrates with Network Node Manager by retrieving nodes from its OpenView database, sending color changes to the node symbols on the map, and sending events to the event browser. The Network Node Manager host requires NerveCenter's OVPA adapter to communicate with NerveCenter.
HMAC-MD5-96	See *MD5*.
HMAC-SHA-96	See *SHA*.

inform message A message indicating a specified network condition. You can configure alarms to send a NerveCenter Inform when a particular network event is detected. NerveCenter can send the Inform data to one or more NerveCenters and network management platforms.

In NerveCenter Administrator, you specify the Inform recipients in the Inform tab and determine which types of events are forwarded (for example, those above a specified severity level). This allows you to manage which platforms handle which events. You can use separate criteria for each platform that is to receive events and coordinate alarms from multiple NerveCenters so that one central NerveCenter determines which events should be forwarded to the platform.

instance The identity of a particular MIB base object when there are multiples of that base object on a node. For example, if a managed node has four ports, it has four instances of the ifEntry base object, numbered one through four.

In NerveCenter, a base object and its instance is called a subobject.

instance scope A setting for an alarm definition that lets you monitor one or more base objects in an alarm instance. Instance scope is similar to Subobject scope, though Instance scope lets you monitor any instance for different base objects.

See also *alarm scope*.

IP address The 32-bit host address defined by the Internet Protocol in STD 5, RFC 791. It is usually represented in dotted decimal notation, for example, 192.164.10.0.

logon rights NerveCenter Client users are grouped into two categories, depending on their logon rights:

- Those with administrator rights can customize and create NerveCenter behavior models and save all changes to the NerveCenter database. These users belong to the NerveCenter Admins group (Windows) or ncadmins (UNIX). They can also log in as an administrator on Windows or root on UNIX.
- Those with user rights belong to the NerveCenter Users (Windows) or ncusers (UNIX) group. They perform such tasks as monitoring NerveCenter alarms, resetting alarms, and generating reports. They cannot modify the NerveCenter database, for example, by making changes to nodes, property groups, polls, masks, or alarms.

managed node A node that can be targeted by NerveCenter polls. Only managed nodes are polled. SNMP traps, however, are received from nodes regardless of their managed state.

See also *node*.

Management information base (MIB)	A defined collection of device information that's governed by SNMP. An agent's MIB contains the configuration and status values for the particular type of device. A specific type or class of management information is called a MIB base object. MIB-II defines the common set of objects for network management of TCP/IP-based intranets. Other enterprise-specific MIBs can be defined to support specific pieces of hardware. See also *Simple Network Management Protocol (SNMP)*.
map	The graphic representation of your network and its devices that a network management platform provides.
mask	See *trap mask*.
MD5	Message authentication/hashing algorithm as defined in HMAC-MD5-96 specifications, intended for secure message exchanges between any two entities communicating with each other. The algorithm takes a message of arbitrary length and produces a 128-bit message digest. This is one of the two algorithms suggested in SNMPv3 specifications for message authentication. This protocol is supported by NerveCenter.
MIB	See *Management information base (MIB)*.
MIB base object	An object that is managed by a network manager. Base objects contain attributes that have values. In MIB-II, for example, system, ifEntry, icmp, and ipAddrEntry are all base objects. System contains attributes such as sysDescr and sysUpTime that have associated values. Some base objects, such as system, are simply the MIB group name and contain a single set of attributes. These are zero-instance base objects because they have no separate instances. Other base objects represent one instance out of two or more. IfEntry, for example, is one interface on a device that may contain several interfaces. Connecting the base object name and instance number with a period, as in ifEntry.2, fully qualifies the base object instance that is being referenced.
IBM Tivoli Netcool/ OMNIbus	A network management platform that can be configured to receive NerveCenter Inform messages. The Netcool platform host requires NerveCenter's universal platform adapter to communicate with NerveCenter.

multitier behavior model	A behavior model that contains two or more alarm definitions. A transition in one alarm instance fires triggers to feed other transitions in other alarm instances. This is useful when you want to correlate conditions that first need to be correlated themselves, or when you want to correlate conditions for a node, but need information from another node to determine the final outcome. This type of behavior model can exploit NerveCenter's alarm scope capability. For example, if you want to be notified when a device has high port traffic, you can design a subobject scope alarm to detect high traffic for a single port. This alarm sends a trigger to a second alarm each time heavy traffic is detected on the port. The second alarm uses node scope and processes at most one trigger for the port. The second alarm notifies you once when the node's port is busy.
NCContext	Every communication between two SNMPv3 entities takes place on behalf of a *user* (a uniquely identified entity in the SNMPv3 management domain) in some *context* (a uniquely identified entity for access control generally configured on conventional SNMPv3 agents/nodes). Before NerveCenter can poll SNMPv3 agents, the agents must be configured to support a NerveCenter user and NerveCenter context. By default, the user is NCUser and the context is NCContext, though you can change both names in NerveCenter Administrator.
NCUser	Every communication between two SNMPv3 entities takes place on behalf of a *user* (a uniquely identified entity in the SNMPv3 management domain) in some *context* (a uniquely identified entity for access control generally configured on conventional SNMPv3 agents/nodes). Before NerveCenter can poll SNMPv3 agents, the agents must be configured to support a NerveCenter user and NerveCenter context. By default, the user is NCUser and the context is NCContext, though you can change both names in NerveCenter Administrator.
NerveCenter Administrator	The NerveCenter software module that enables you to customize and manage NerveCenter operations. Run the NerveCenter Administrator program, ncadmin.exe, from your Start menu program group (Windows NT) or from the NerveCenter_*installation*/bin directory.
NerveCenter administrator	A system or network administrator who is responsible for configuring and maintaining NerveCenter operations and, optionally, installing NerveCenter.
NerveCenter Client	The NerveCenter software module that enables you to create and manage behavior models as well as monitor and report on network activity. Run the NerveCenter Client program, client.exe, from your Start menu program group (Windows NT) or from the NerveCenter_*installation*/bin directory.
NerveCenter enterprise trap number	The enterprise-specific MIB object identifier (OID) for NerveCenter is 1.3.6.1.4.1.78, used to create trap masks that detect Inform messages.

NerveCenter object	A NerveCenter entity that can be part of a behavior model used to manage your network. The following are NerveCenter objects: nodes, property groups and their properties, polls, trap masks, and alarms.
NerveCenter Server	The background engine that handles event correlation and manages the NerveCenter database. Each time you use the NerveCenter Client or Administrator, you must connect to a server. The server can be installed on the same machine as the NerveCenter Client or Administrator applications.
NerveCenter Web Client	The NerveCenter software module that enables you to monitor network activity using Microsoft Internet Explorer 4.0 or Netscape Navigator 4.0. Run the Web Client from your browser by entering the following address: `http://servername/NerveCenter` See also *NerveCenter Web Collector*.
NerveCenter Web Collector	If Web server integration was included with your NerveCenter installation, the NerveCenter Web Collector was installed on your machine. The Web Collector communicates with the NerveCenter Server and your Web server to provide current information about alarms associated with the NerveCenter Server.
network management platform	NerveCenter supports several network management platforms, and communicates with each using a platform adapter. The platform adapter must be installed and running on the platform host before a connection can be established with NerveCenter. Hewlett Packard OpenView Network Node Manager can both provide node information and receive NerveCenter messages using NerveCenter's OVPA adapter. IBM Tivoli Netcool/OMNIbus can only receive NerveCenter messages and display those messages in their event or message browser. It uses the NerveCenter universal platform adapter:
Network Node Manager	See *Hewlett Packard OpenView Network Node Manager*.
NoAuthNoPriv	Security level that does not require message authentication or encryption services for communication between any two SNMPv3 entities. That means communication between the two SNMPv3 entities communicating with this (i.e. NoAuthNoPriv) security level is not secure. Communication between two SNMPv3 entities takes place on behalf of a *user* (a uniquely identified entity in the SNMPv3 management domain), and possibly in some *context* (a uniquely identified entity for access control generally configured on conventional SNMPv3 agents/nodes).

node	Any device on the network that can be managed. Examples of nodes are servers, workstations, printers, hubs, routers, bridges, and gateways. When NerveCenter is integrated with a network management platform, you can configure NerveCenter's node list to be populated by that platform. In addition, NerveCenter can be configured to add unknown nodes that send it a trap. You can also populate the list manually or by using the Discovery behavior model. See also *managed node*.
node scope	Alarm definition setting that determines if each alarm instance monitors a node. See also *alarm scope*.
object identifier, object ID, or OID	A unique SNMP name given to each MIB base object, identified by the value of the sysObjectID object in the system group of MIB-II. The name is written as a sequence of integers separated by periods. For example, the sequence 1.3.6.1.2.1.1.1.0 specifies a system description, where 1.3.6.1.2.1.1 specifies the base object system, 1.3.6.1.2.1.1.1 specifies the attribute sysDescr (which has a value), and 1.3.6.1.2.1.1.1.0 specifies an instance (0) Objects associated with a vendor or equipment type also have object identifiers. You can obtain the object identifier for a network node by opening its Node Definition window. Select the Query Node tab and then select the **Get** button.
OID	See *object identifier, object ID, or OID*.
OVPA (OpenView Platform Adapter)	A NerveCenter platform adapter process that resides on a Hewlett Packard OpenView Network Node Manager. The adapter enables communication between the network management platform and NerveCenter.
passwords (SNMPv3)	Before NerveCenter can poll SNMPv3 nodes, you must supply the proper passwords for the NerveCenter user communicating with that node. These passwords must be configured on your SNMPv3 agents as well as in NerveCenter Administrator. You need to provide the following passwords: **Authentication -** Required for providing authentication services when NerveCenter communicates with the agent using either an AuthNoPriv or AuthPriv security level. **Privacy -** Required for providing encryption services when NerveCenter communicates with the agent using an AuthPriv security level. Depending on the security level you provide in NerveCenter Client for the node corresponding to an SNMPv3 agent, NerveCenter will use an appropriate combination of passwords to provide message authentication and message encryption services. No passwords are required if using NoAuthNoPrive security.

ping	An ICMP echo request sent to a network device that returns an echo reply from the device. ICMP is an error-control protocol.
platform adapter	A NerveCenter process that facilitates communication between NerveCenter and a network management platform. The OVPA adapter enables NerveCenter to receive node information from and send messages to that platform. The universal platform adapter enables NerveCenter to send messages to a platform. See also *OVPA (OpenView Platform Adapter)* and *Universal Platform Adapter (paserver)*.
poll	A NerveCenter object that monitors the network for conditions of interest by polling SNMP agents on managed nodes for specific MIB data. These polls are compare to a user-defined poll condition and trigger expression. If the condition is satisfied the poll generates a trigger that signals one or more alarms, at least one of which must be enabled and include a transition that the poll can trigger.
poll condition	Defines the condition to be detected on each node that is tracked. When a poll condition is satisfied, the poll generates a trigger that signals one or more alarms. You can create conditions, such as threshold crossings or rates of change for tracked values, using arithmetic and relational operators. You can string together combinations of conditions using logical (Boolean) operators.
poll rate	The interval (in seconds, minutes, or hours) that a poll waits between requests for node MIB data and evaluation of the associated poll condition.
principal	The person who has access to an SNMPv3 agent.
privacy key	The seed used by the privacy protocol to encrypt messages while communicating with an SNMPv3 entity using AuthPriv security level. The privacy key is described in SNMPv3 specifications.
property	A text string that is a member of one or more property groups. Properties fall into two categories: MIB base objects and user-defined strings. Poll and alarm definitions use properties when determining whether a device should be monitored. The property group associated with a managed node must contain the properties for every MIB base object used in a poll for that node. The property groups shipped with NerveCenter contain the base objects relevant for the group. For example, the Mib-II property group contains properties for each base object in MIB-II. User-defined strings are used to restrict the targeted subset of nodes for a poll or alarm.

property group

A collection of one or more text strings called properties. Property groups allow you to categorize nodes into logical or managerial units. The groups can be based on device type, location, priority, supported MIBs, business function, or any other useful characteristic.

Each managed node belongs to a property group and, therefore, is associated with all of the group's properties. These properties are used to restrict which polls and alarms monitor the node. Assigning a node to a property group that contains multiple properties allows the node to be targeted by multiple behavior models.

resynchronization (resynch)

When NerveCenter is integrated with a network management platform, NerveCenter issues a Resync command upon start-up. This command updates NerveCenter's entire node list with information from the platform. The update contains all nodes within NerveCenter's subnet filters, if applicable, and includes details about each node's parents.

You can manually refresh the NerveCenter node list by selecting the **Resync** command from the client's Server menu. If this command is not available, the connection to the node source is down.

See also *synchronization*.

scope

See *alarm scope*.

security level

The use of message authentication and message encryption services for communication between any two SNMPv3 entities. NerveCenter supports the following security levels while communicating with SNMPv3 agents:

NoAuthNoPriv - Neither message authentication nor message encryption services are used for communication between two SNMPv3 entities. As such, the communication is not secure.

AuthNoPriv - Message authentication services are used without message encryption for communication between two SNMPv3 entities. This ensures only authenticity of the messages being exchanged between two SNMPv3 entities. The messages are not encrypted.

AuthPriv - Both message authentication and message encryption services are used for communication between two SNMPv3 entities. This is the most secure way of communication defined in SNMPv3 specifications at the present time.

SerializeDB

A utility that transfers data between the database being used by NerveCenter and a serialized file. This tool is useful for backing up and restoring database information. Run the SerializeDB program, serializedb.exe, from your Start menu (Windows NT) or from the NerveCenter installation/Bin directory.

See also *DBWizard*.

severity	See *alarm severity*.
SHA	A secure hash algorithm specified in the Secure Hash Standard (SHS, FIPS PUB 180). This is one of the two algorithms suggested in SNMPv3 specifications for message authentication. This protocol is supported by NerveCenter.
Simple Network Management Protocol (SNMP)	An industry-standard protocol that defines how network management systems exchange information with their managed nodes. Software processes called SNMP agents reside on each managed device and track defined sets of data. A database of network information, called a management information base (MIB), is associated with both the manager and agent. A manager and agent may exist on the same system. SNMP messages allow a management application to set and retrieve agents' managed data. SNMP traps allow an agent to send events to a management application.
smart polling	A NerveCenter feature that reduces the network overhead associated with SNMP management. A poll doesn't solicit data from nodes unless an alarm definition uses the poll's trigger. Even then, the poll is issued to a specific node only when the alarm exists in a state that the poll can trigger. For example, if a behavior model correlates high traffic followed by high error rates, a node is not polled for error rates unless it fulfills the high traffic condition first.
SNMP	See *Simple Network Management Protocol (SNMP)*.
SNMP trap	An asynchronous notification that an SNMP agent can send to a management application. Traps are identified by a generic number (0–6), an enterprise-specific number, and an enterprise name. Related information data is typically bundled with a trap in structured formats called variable bindings. NerveCenter detects and filters SNMP traps with trap masks.
specific trap number	An SNMP trap number associated with an enterprise MIB. When defining trap masks, generic trap numbers (0-6) indicate the nature of the SNMP trap being reported. In addition, each enterprise, or vendor, can have any number of subtrap numbers, called specific trap numbers, within the category 6 generic trap. Each vendor defines its own number of enterprise traps, their associated specific numbers, their functions, and their variable bindings. See also *generic trap number*.
state diagram	See *alarm state diagram*.
subobject	A base object and an instance connected by a period. Examples from MIB-II include ifEntry.3, system.0, and ipForwardEntry.2.

subobject scope	A setting for an alarm definition that determines that each alarm instance monitors a subobject. Subobject scope indicates that conditions for each subobject are tracked separately with separate alarm instances. For example, one alarm instance would track conditions on ifEntry.1, another would track conditions on ifEntry.2, and so on. See also *alarm scope*.
suppression	NerveCenter provides the following types of suppression as a way to manage unresponsive nodes. **Node suppression** - When a node fails to respond, it can be suppressed manually or through an alarm action. Normal polling does not occur for the node as long as the node remains suppressed. **Poll suppression** - By default, NerveCenter polls are suppressible. This means they do not poll nodes that are suppressed. However, there may be specific polls that you want to occur on a node even when it is suppressed. In such a case, you can set the poll to insuppressible. Suppression affects only polling; SNMP traps are still received from a managed node even though it is suppressed.
synchronization	When NerveCenter is integrated with a network management platform, the platform uses a process called synchronization to update NerveCenter whenever the platform adds, updates, or deletes nodes. This type of update does not include parent information. See also *resynchronization (resynch)*.
time ticks	See *engine time ticks*.
transition	See *alarm transition*.
trap	See *SNMP trap*.
trap mask	A NerveCenter object that captures SNMP traps received from managed nodes and fires a trigger that transitions an alarm instance. A mask can detect a standard SNMP trap identified by one of six generic categories, an enterprise-specific trap, and, with the advanced Trigger Function feature, a trap whose contents match user-specified criteria. At least one alarm must be enabled and include a transition that can be triggered by the mask’s trigger.
trigger	A flag sent to one or more alarms to indicate that an event has been detected or a condition satisfied. The trigger transitions an alarm from one state to another. The alarm must be enabled and include a transition that can be triggered by this particular trigger.

	Triggers are generated by polls, trap masks, and the Fire Trigger or Perl Subroutine alarm actions. You specify a trigger using the FireTrigger() function in a poll condition, Perl subroutine, or trap mask trigger function. NerveCenter generates its own built-in triggers, designed to detect unresponsive nodes.
trigger function	A Perl script that you define for a NerveCenter trap mask. The script is called whenever a detected SNMP trap matches the mask's data fields. The Perl script issues triggers based upon the contents of the trap's variable bindings. When a suitable trap is detected, NerveCenter executes the trigger function, supplying data from the trap's variable bindings. Your script evaluates the variable-binding contents and conditionally fires triggers or assigns property groups.
trigger generator	A NerveCenter object or action that fires a trigger when a certain condition is detected. The trigger is sent to one or more alarms, where it transitions one or more alarm states. The following are NerveCenter trigger generators: polls, trap masks, Perl subroutines that include the FireTrigger() function, and alarm actions (Send Trap, Fire Trigger). NerveCenter generates its own built-in triggers when it detects unresponsive nodes.
Universal Platform Adapter (paserver)	A NerveCenter platform adapter process that resides on IBM Tivoli Netcool/ OMNIbus. The universal platform adapter enables NerveCenter to send Inform messages.
user rights	Those with user rights belong to the NerveCenter Users (Windows NT) or ncusers (UNIX) group. They perform such tasks as monitoring NerveCenter alarms, and resetting alarms. They cannot modify the NerveCenter database, for example, by making changes to nodes, property groups, polls, masks, or alarms.
variable bindings	An array of related information that accompanies an SNMP message. The variable bindings are typically defined in standard or vendor-specific MIB definitions (.ASN1 files). The format of each variable binding is defined by SNMP. NerveCenter polls and trap masks obtain variable bindings from the nodes they monitor.
Zero-instance base object	A MIB base object that has only one instance and, therefore, contains a single set of attributes. For example, system is a zero-instance base object because it contains attributes (like sysLocation) that have only one associated value per node. In contrast, ifEntry is a base object with multiple instances. If a node had four ports, the node would has four instances of the ifEntry base object, numbered one through four, and each instance has associated attributes.

Index

A

Action Manager error messages 239
Action Router 18
Action Router alarm action 95
Action Router tool 95
activating 106
Active Directory 174
active NerveCenter Server 43
add node command, nccmd 322
administrator
  role 36
Administrator, NerveCenter 24
agents
  SNMP v3 144
alarm 15, 18
  finite state machine 18
  IfUpDownStatus example 18
Alarm Action counters 265
alarm actions 18, 179
  Action Router 95
  FireTrigger 94
  troubleshooting 194
Alarm Filter error messages 243
Alarm Summary window 25
ASN.1 files
  capitalization 272
  characters 271
  troubleshooting 271
authentication 147
auto-classification
  how NerveCenter classifies nodes 133
  maximum classify value 137

B

behavior models 14
  objects 15
  operation 16

C

capabilities, filtering by 97
CLI 28
client command 283
Client, NerveCenter 26
command line interface 28, 278
commands
  client 283
  dbwizard 283
  importutil 284
  installdb 293
  ipsweep 294
  mibcomp 296
  mibTool 298
  ncadmin 299
  nccmd 300
  nclicreport 301
  ncserver 301
  ncstart 303
  ncstop 304
  ovpa 306
  paserver 310
  serializedb 313
  trapgen 314
communication ports 74, 80
conditions
  finding sequences 90
  finding set of network 89
  network, detecting 87
  persistent network 88
  responding to network 92
Connect to Server window 43
Connecting to a server
  troubleshooting 50
corrective actions 94
correlating conditions 87
Correlation 13
counters
  Alarm Action 265
  Database 265
  Inform 266
  Inform NerveCenter 266
  Log to Database 267
  Log to File 267
  Node Source 268
  Poll 269
  refreshing 264
  resetting 263
  Server 268
  trace 263

D

Database counters 265
database management
  DBWizard 206
  SerializeDB (UNIX) 207
  UNIX 207
  Windows 208
database, NerveCenter 21
DBWizard 206
dbwizard command 283
defining
  nodes 86
Deserialize Manager error messages 243
detecting condition persistence 88
detecting conditions 87
digest keys 129, 147
disabling 106
Discover Nodes from Traps 118
documentation
  conventions 6
  feedback 7

E

environments
  UNIX 38
error messages
  Action Manager 239

Alarm Filter 243
Deserialize Manager 243
Flatfile Managerr 243
Inform OV Manager 244
Inform Product Manager 244
LogToDatabase 246
LogToFile Manager 247
OpenView configuration 257
PA Resync Manager 248
Poll Manager 247
Protocol Manager 247
Server Manager 250
Trap Manager 254
UNIX installations 255
user interface 236
error status
SNMP v3 operations 153
event monitoring
ICMP 162
events 13

F

finding set of network conditions 89
finite state machine 18
FireTrigger alarm action 94
flat files in UNIX 204
Flatfile Manager error messages 243

G

Glossary 331

H

hostname filter 104, 106
hostname, filtering by 104

I

ICMP 12
event monitoring 162
IIS, configuring 198
importing 61
ImportUtil 61
troubleshooting 71
importutil command 284
imputil.ini 62
troubleshooting 71
inform 18
troubleshooting 83
inform actions
configuring 78
Inform counters 266
Inform NerveCenter counters 266
Inform OV Manager error messages 244
Inform Product Manager error messages 244
installdb command 293
integration with network management platforms 32
Internet Control Message Protocol 12
IP filter 99
activating 106
enabling 106
IP filter, disabling 106
IP filters
troubleshooting 121
IPSweep behavior model 112
ipsweep command 294

K

keys, digest 129

L

License tab 141
log management 191
Log to database alarm action 191
Log to Database counters 267
Log to File alarm action 191
Log to File counters 267
logging 93
logging NerveCenter activity 262
login rights 34, 35
logs
settings 191
SNMP v3 operations 149
troubleshooting 194
LogToDatabase Manager error messages 246
LogToFile Manager error messages 247

M

mail actions
MS Mail 182
SMTP mail 180
management information base 22
maximum classify value 137
mibcomp command 296
MIBs
adding definiitons 228
adding or modifying 228
compiling 231
mibcomp compiler 22
NerveCenter definitions 226
source definitions 226
troubleshooting 233
troubleshooting ASN.1 files 271
vendor-specific 227
mibTool command 298
Microsoft Access 204
Microsoft Exchange profiles 182
Microsoft mail 182
troubleshooting 194
Microsoft SQL Server 204
MS Mail 182
multiple servers 44
multi-threaded polling 140

N

ncadmin command 299
NCAdmin Configuration window 45
nccmd
add node command 322
object commands 322
server commands 319
nccmd command 300
nclicreport command 301
ncserver command 301
ncstart command 303
ncstop command 304
NerveCenter
Action Router tool 95
administration 33
Administrator 24
Client 26
components 21
database 21
inform actions 78
managing nodes 14, 86
Server 21
servers, multiple 31
NerveCenter Administrator

connecting to a NerveCenter Server 43
server port 76
starting in UNIX 42
starting in Windows 42
NerveCenter Client
Alarm Summary window 25
login rights 35
NerveCenter Client console 25
NerveCenter database 205
backing up 209
formats 204
restoring 213
transferring data 218
troubleshooting 224
using the command line 216
NerveCenter installation error messages (UNIX) 255
NerveCenter MIB
troubleshooting ASN.1 files 271
NerveCenter OpenView Platform Adapter (OVPA) 110
NerveCenter Server
active 43
as a process 186
as a Windows service 183
as an Inform recipient 78
configuring from the command line 61
connecting 43
connecting to multiple 44
connection port 74
disconnecting 49
starting on UNIX 38
starting on Windows 39
status 46
stopping on UNIX 38
stopping on Windows 39
troubleshooting 40, 71
NerveCenter utilities
ImportUtil 61
Trapgen 53
Traprcv 54
NerveCenter Web Client 27
NerveCenter Web Collector
on UNIX 196
on Windows 196
network conditions
detecting 87
finding set of 89
persistent 88
responding to 92
network management platform
as a node source 115
node source filters 97
resynchronization 116
synchronization with 115
troubleshooting 121
network management strategy 29
node data source 110
node list 96
adding and deleting manually 119
filtering 97
filtering by capabilities 97
filtering by hostname 104
filtering by IP address 99
filtering by system object identifier 98
maintaining 115
populating 109
populating manually 114
populating using a network management platform 110
populating using the IPSweep behavior model 112
troubleshooting 121
node source - server status
processing unknown traps 107
Node Source counters 268
nodes 15
autodelete 119
classification algorithm 131
classification tuning 138
classifying 130, 132
data sources 96
defining 86
deleting manually 119
discovered from traps 117
hostname filter 104
managing 14
renaming 117
version classification 133
notification 93
notification action
MS Mail 182
page (UNIX) 189
page (Windows) 188
SMTP mail 180

O
object commands, CLI 322
object ID 98
online knowledgebase 8
OpenView configuration error messages 257
operations log 149
signing for errors 150
viewing 152
ovpa command 306

P
PA Resync Manager error messages 248
page actions 189
paging
troubleshooting 194
UNIX notifications 189
Windows notifications 188
PAM 166
paserver command 310
passwords 147
SNMP v3 129
Perl 18
poll 15
Poll counters 269
Poll Manager error messages 247
polling
multi-threaded 140
ports
SNMP settings 143
privacy 147
property 15
property group 15
Protocol Manager error messages 247

R
renaming nodes 117
resetting counters 263
responding to network conditions 92
role
administrator 36

S
security
SNMP v3 128

I Index

- UNIX 166
- Windows 170

SerializeDB
- application 213
- troubleshooting 224
- UNIX 207

SerializeDB application 210
serializedb command 313
server commands, CLI 319
Server counters 268
server list
- changing ports 76
- deleting names 45

Server Manager error messages 250
server ports
- CLI 80
- server communication 74
- SNMP 81

server status
- CLI port 80
- configuring the Administrator port 76
- configuring the Server port 74
- license information 141
- managing the server list 45

Server Status window 46
servers, multiple NerveCenter 31
severity 25
Simple Network Management Protocol 12
SMTP
- mail notifications 180
- troubleshooting notifications 194

SNMP 12, 13
- port settings 143
- ports 81
- trap support 53
- version classification 133

SNMP Agent
- disabling 159

SNMP Agent tab 157
SNMP master agent
- configuring 157

SNMP pollers 140
SNMP settings
- security 147

SNMP sub agent
- configuring 160

SNMP trap source 52
SNMP v3
- auto-classification 137
- configuring security 147
- configuring trap source 55
- digest keys 129
- discovering and initializing agents 144
- error status 153
- initial user 146
- manual classification 137
- operations log 149
- passwords 129
- security 128
- security configuration 147
- security levels 128
- security support 128
- support overview 126

SNMP v3 support
- trap source 52

standalone operation 30
state transitions *See* transitions
status, SNMP v3 operations 153
subnet filter
- automatic criteria 101
- manual criteria 102

system object ID 98
system object identifier, filtering by 98
system Object Identifiers 110

T

Tech Support Web 36
technical support 7
- contacting 8
- educational services 8
- professional services 7

trace counters
- using 263

tracing NerveCenter operations 262
transitions
- causing 94

trap
- processing unknown nodes 107

Trap Manager error messages 254
trap mask 15
trap source 52
Trapgen 53
trapgen command 314
Traprcv 54
traps
- SNMP support 53

traps from unknown nodes
- processing 107

trigger 15

U

UNIX
- database 207
- managing security 166
- paging 189
- troubleshooting 50

unknown nodes 117
user interface messages 236
user types 34

W

Web Client 27
Web Collector 27
- configuring 197
- starting and stopping 196

web integration 195
Web-based monitoring 28
Windows
- database 208
- domain controller security 172
- managing security 170
- paging 188
- troubleshooting 40
- workstation/server security 170