Open NerveCenter™ 3.8

Managing NerveCenter

UNIX and Windows

July 2003

Disclaimer

The information contained in this publication is subject to change without notice. OpenService, Inc. makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. OpenService, Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this manual.

Copyright

Printed in the USA, July 2003.

Open NerveCenter *Managing NerveCenter*

OpenService, Inc.
110 Turnpike Rd., Suite 308
Westborough, MA 01581
Phone 508-366-0804
Fax 508-366-0814
http://www.open.com

Contents

Understanding NerveCenter 1

This chapter explains:

- What type of product NerveCenter™ is
- How NerveCenter manages nodes
- What the NerveCenter main components are
- What roles NerveCenter can play in a network or system management solution

For information on these topics, see the sections shown in the table below.

Table 1-1. Sections Included in this Chapter

Section	Description
What is NerveCenter? on page 2	Explains that NerveCenter is an advanced event automation solution.
How NerveCenter Manages Nodes on page 3	Explains how NerveCenter isolates and responds to emerging network and system problems.
Main NerveCenter Components on page 13	Discusses NerveCenter's client/server architecture. Explains how NerveCenter tracks network conditions using finite state machines called alarms, where these alarms get their input, and how alarm transitions can result in actions.
Role in Network Management Strategy on page 21	Explains how NerveCenter can be used stand-alone, integrated with other NerveCenter systems, or integrated with other Open or third-party products.

What is NerveCenter?

As corporations have focused attention on keeping their corporate networks available at all times, they have invested heavily not only in redundant hardware, but also in network management software. Unfortunately, many network management tools whose purpose is to identify network faults can overwhelm operators with raw network data. Only after manually sifting through this raw data and identifying the real problems can operators take the appropriate corrective actions.

NerveCenter is different. It is able to isolate and respond to network conditions proactively. In addition, NerveCenter is a highly-scalable, cross-platform solution.

At the heart of NerveCenter is its event correlation engine. For each device that it is monitoring, NerveCenter creates one or more finite state machines—or alarms—that define operational states it wants to detect. NerveCenter also defines rules that effect transitions between the 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.

These state machines enable NerveCenter to correlate data from multiple sources over time before it concludes that a problem exists. As a simple example, if NerveCenter receives a link-down trap for an interface, it does not immediately report a problem; instead, it waits for a link-up trap for that interface. If NerveCenter receives a link-up trap within a given amount of time, it can ignore both traps. Otherwise, it can report that a particular communication link is down.

Once NerveCenter has identified a problem, it can take automatic corrective actions. A variety of actions can be associated with state transitions, including notifying an administrator, executing a program or script that corrects the problem, or notifying a network management platform of the network condition.

In addition to being an advanced event automation solution, NerveCenter is also a highly scalable 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 be distributed as a background process at tens or even hundreds of remote offices.

Finally, NerveCenter is a cross-platform solution. NerveCenter automatically correlates events, identifies problems, and takes corrective actions across network devices running an SNMP agent, UNIX systems, and Windows workstations and servers. The capability for NerveCenter components on Windows systems to work with components on UNIX systems enables you to install NerveCenter on the type of system—hardware and operating system—that is most appropriate for a job. For instance you might install NerveCenter on a Windows system to monitor a small network of 1000 nodes or fewer, and you might install NerveCenter on a symmetric multiprocessor UNIX server to manage several thousand nodes. You could monitor and configure both of these systems from a Windows or UNIX workstation.

How NerveCenter Manages Nodes

To perform its job of event automation, NerveCenter relies on the definition of *behavior models*. These models are constructed from NerveCenter objects (which we'll discuss in detail later) and define:

- Which nodes the 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

The following sections elaborate on the tasks that NerveCenter performs in order to automate event handling:

- *Defining a Set of Nodes* on page 3
- *Detecting Conditions* on page 4
- *Correlating Conditions* on page 4
- *Responding to Conditions* on page 9

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

As is discussed in the section *Role in Network Management Strategy* on page 21, NerveCenter can collect network and system data from a variety of sources. However, most frequently NerveCenter 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 SNMP v2c and v3 features allow SNMP to be utilized both reliably and securely.

Correlating Conditions

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

- How these conditions, or some subset of them, are related
- The underlying cause of a set of conditions, 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:

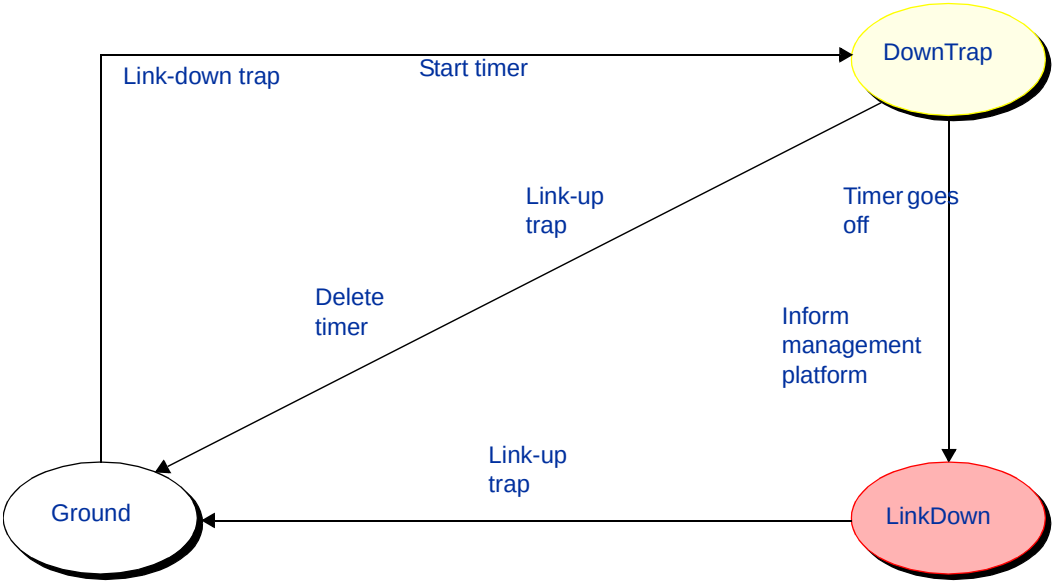
- *Detecting the Persistence of a Condition* on page 5
- *Finding a Set of Conditions* on page 6

- *Looking for a Sequence of Conditions* on page 7

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 below.

Figure 1-1. State Diagram for Detecting a Link-Down Condition



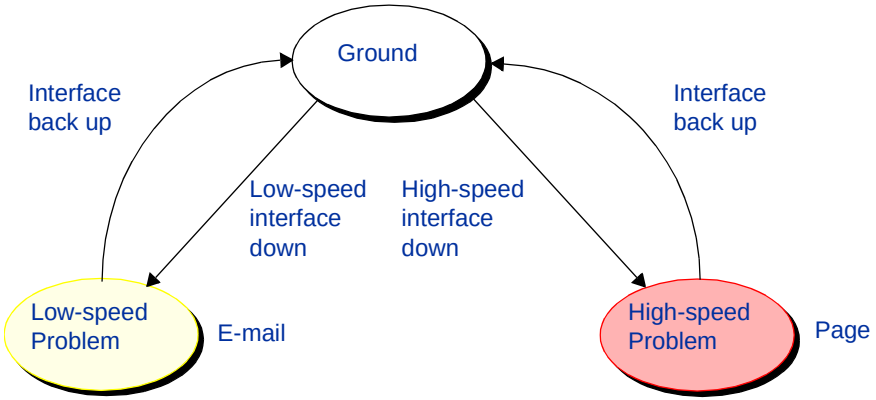
Let's say that NerveCenter has this state diagram in memory and is tracking a particular interface for a link-down condition.

- 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 the identification of a set of conditions. For example, let's say that you're monitoring the interfaces on a router. To be notified when a low-speed interface goes down or when a high-speed interface goes down, you might use the following state diagram.

Figure 1-2. 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 this set of conditions is met, a problem exists on an interface that is probably used for a dial-up connection.

The second set of conditions is the same as the first, except that the last condition is that the interface's current bandwidth is greater than or equal to 56K. If this 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.

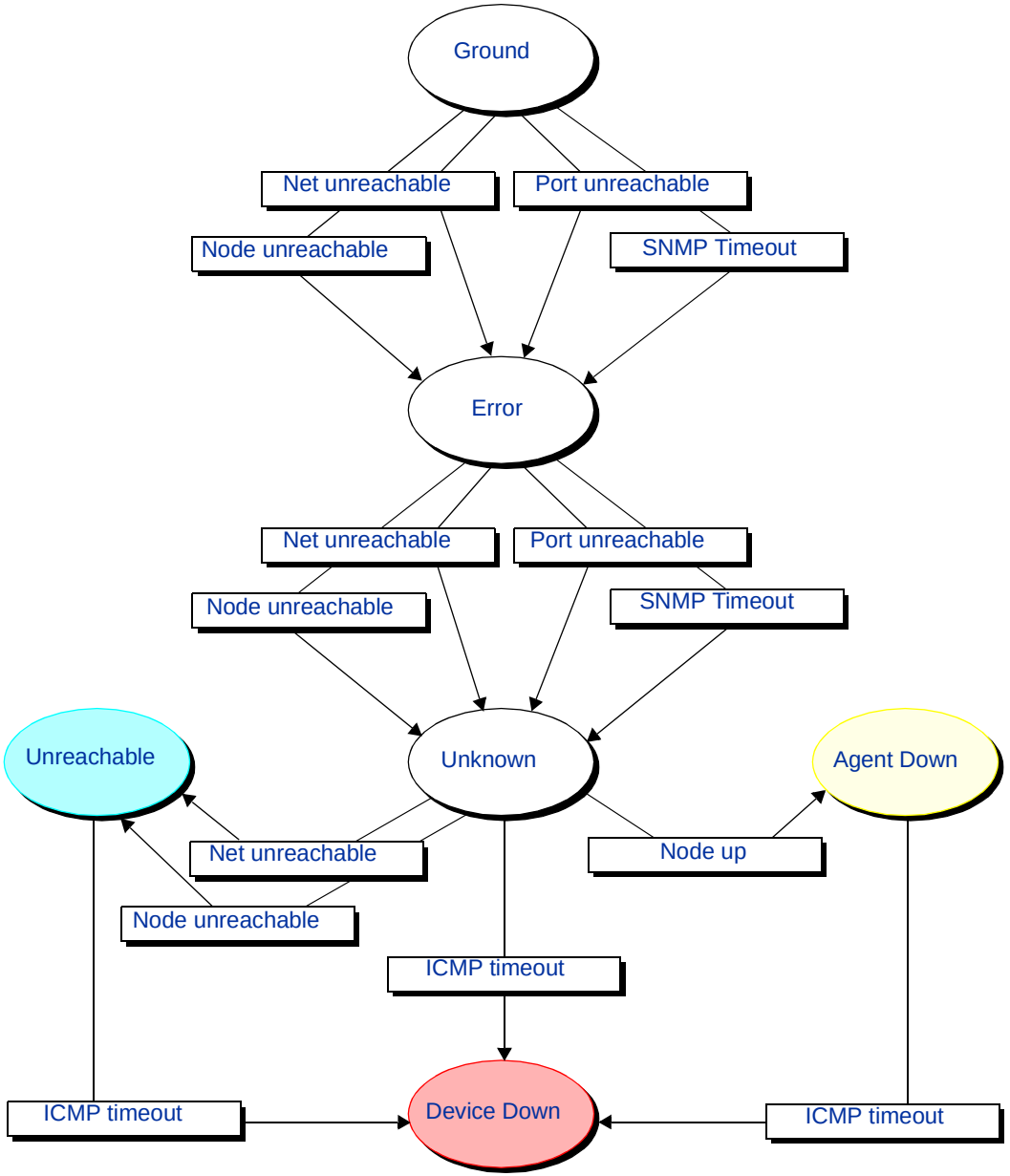
NerveCenter may detect many conditions concerning an interface before it finds the set of conditions it is looking for. The administrator need not see information about each of these conditions. He or she will be emailed or paged if the interface goes down.

Looking for a Sequence of Conditions

NerveCenter also enables you to correlate conditions by looking for sequences of conditions. This type of correlation is possible because, in NerveCenter, 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 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 1-3. 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 10
- *Logging* on page 10
- *Causing State Transitions* on page 11
- *Corrective Actions* on page 11
- *Action Router* on page 12

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 Micromuse'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 21.

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 that processes 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 any 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 an IT/Operations message
- The contents of the varbind data associated with a trap or a poll

Main NerveCenter Components

NerveCenter is a distributed client/server application and includes the following components:

- Server
- Database
- Clients

For information about these components, see the following sections:

- *The NerveCenter Server* on page 13
- *The NerveCenter Database* on page 13
- *The NerveCenter User Interface* on page 17

The 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.

The NerveCenter Database

The NerveCenter database is primarily a repository for the NerveCenter objects that make up a set of behavior models. The principal objects used in these models are:

- Nodes
- Property groups and properties
- Polls
- Trap masks
- Alarms

For brief explanations of what these objects are and how they are used, see *Objects in the Database* on page 14.

A set of objects that define many useful behavior models ships with NerveCenter and is available as soon as you've installed the product. For a list of these predefined behavior models, see the section *Predefined Behavior Models* on page 16.

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.

Objects in the Database

This section contains brief definitions of the basic objects used in the construction of behavior models.

- Nodes

  A node represents either a workstation or a network device, such as a router. Each node has an attribute called its property group that controls which behavior models NerveCenter will employ in managing the node.

Note Strictly speaking, a node is not part of a behavior model; rather, it is the entity managed by a behavior model.

- Property groups and properties

  As mentioned above, each node has a property group. This property group is simply a container for a set of properties, which are strings that typically either describe the type of node or name an object in the MIB used to manage the node. It is actually a node's properties, rather than its property group, that determine whether a particular behavior model will be used to manage that node.

- Polls

  A poll defines what MIB variables NerveCenter should request the values of, how those values should be evaluated, and what action the poll should take. If the poll takes an action, it will be to fire a *trigger*, which may cause a state transition in one of NerveCenter's finite state machines.

- Trap masks

  A trap mask describes an SNMP trap and contains the name of a trigger. If NerveCenter receives an SNMP trap that matches the description given in the trap mask, NerveCenter fires a trigger with the name defined in the trap mask. If NerveCenter receives a trap that does not match a trap mask, it discards that trap.

- Alarms

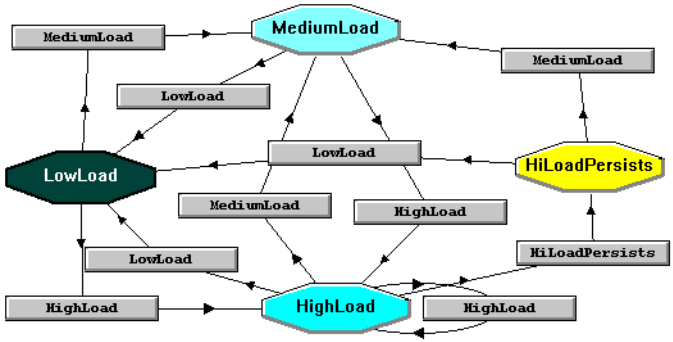
  NerveCenter's finite state machines are called *alarms*. Each alarm defines a set of operational states (such as Normal and Down) and transitions between the states. Transitions are effected by the receipt of the proper trigger and can have actions associated with them. If actions are associated with a transition, the server performs these actions each time the transition takes place.

Behavior Models

Once a set of managed nodes has been defined, NerveCenter's monitoring activities are controlled by a set of *behavior models*. A behavior model is the group of NerveCenter objects required to detect and take action upon a single network condition, such as high traffic on an interface.

The central object in each behavior model is a deterministic finite state machine called an *alarm*. For instance, the alarm shown in Figure 1-4 tracks the level of traffic on an interface.

Figure 1-4. Alarm State Diagram



The possible states in this alarm are low, medium, and high. And these states have the *severities* Normal, Medium, and High, respectively. (The color of each state denotes its severity.) The gray rectangles in the alarm represent *state transitions*.

What about the inputs and outputs of the state machine? The inputs are called *triggers* and can come from several sources. For example, one predefined NerveCenter poll queries the SNMP agent on a device for the level of traffic on, and the capacity of, each interface on the device. If the level of use exceeds a certain percentage of the capacity for an interface, the poll fires the trigger mediumLoad, which can cause a state transition in an alarm.

The outputs of an alarm are called *alarm actions*. These actions are associated with the transition from one state to another by the designer of a behavior model, and NerveCenter performs these actions each time the transition occurs. There are many possible actions, including the following:

- Sending an audible alert to the workstation on which the NerveCenter Client is being run
- Executing a program or script
- Deleting a node from the NerveCenter database
- Informing a network management platform of a condition
- Logging information to a disk file
- Sending mail to an administrator
- Paging an administrator

- Sending an SNMP trap
- Setting a MIB attribute

Predefined Behavior Models

When you install NerveCenter and create a new database, that database contains the objects that make up a number of predefined behavior models. These include behavior models for:

- Detecting authentication failures
- Monitoring the error rate on network interfaces
- Monitoring link-up and link-down traps
- Monitoring the amount of traffic on network interfaces
- Indicating the status of network interfaces: up, down, and so on
- Detecting errors that inhibit accurate SNMP device management
- Determining whether a device is down, unreachable, up without an agent, or up with an agent
- Giving early warning concerning TCP connection saturation
- Verifying that the current TCP retransmission algorithm is the most efficient
- Categorizing devices based on TCP retransmission activity
- Logging information about SNMP traps

NerveCenter also includes predefined behavior models that you can import to monitor specific vendors' devices and additional models for troubleshooting, interface status, data collection, and downstream alarm suppression. For more information about behavior models, see *Designing and Managing Behavior Models*.

The NerveCenter User Interface

The principal clients of the NerveCenter server are:

- The NerveCenter Administrator
- The NerveCenter Client
- The NerveCenter Web Client
- The NerveCenter command line interface

The NerveCenter Administrator is used to configure NerveCenter once it has been installed. The NerveCenter Client and the NerveCenter Web Client are used to monitor a network for problems. The NerveCenter Client is also used to create new behavior models. The command line interface can be used to perform a limited number of operations on NerveCenter objects.

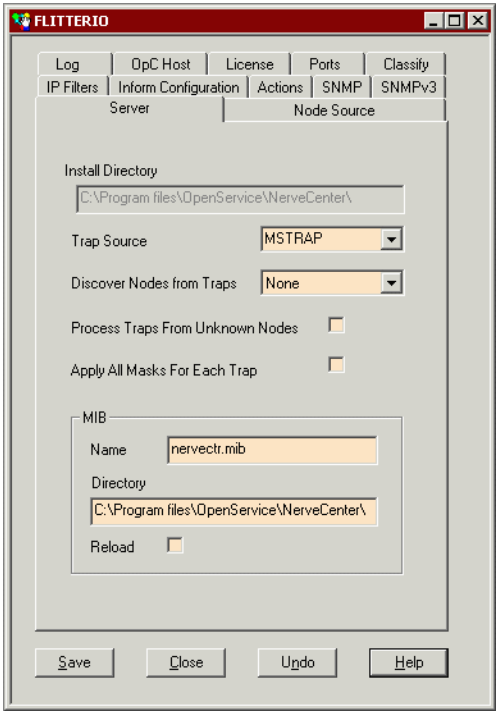
For additional information on these interfaces, see the following sections:

- *The NerveCenter Administrator* on page 18
- *The NerveCenter Client* on page 19
- *The NerveCenter Web Client* on page 20
- *The Command Line Interface* on page 20

The NerveCenter Administrator

Figure 1-5 shows the graphical user interface (GUI) for the NerveCenter Administrator.

Figure 1-5. NerveCenter Administrator



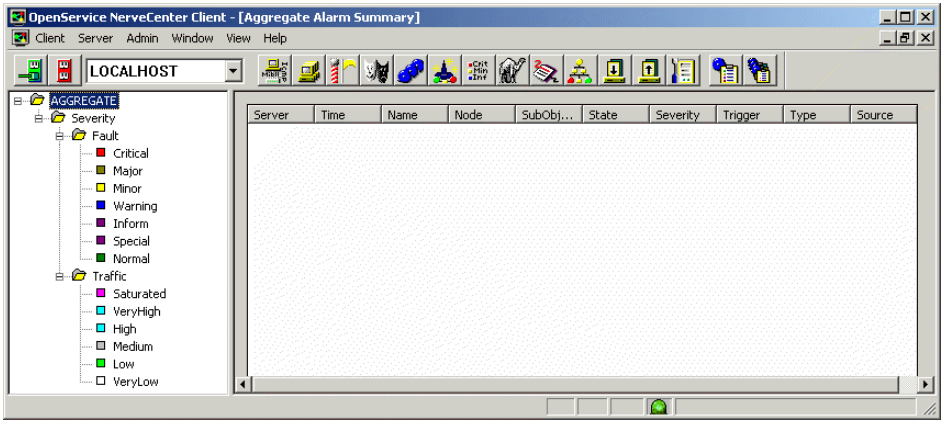
Users with NerveCenter Administrator privileges can use this interface to:

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

The NerveCenter Client

The figure below shows the GUI for the NerveCenter Client.

Figure 1-6. 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 the book *Monitoring Your Network.*

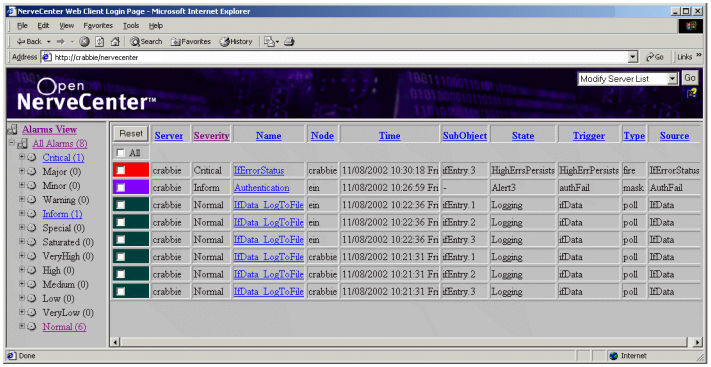
Users with NerveCenter Administrator privileges can perform all the tasks that users with User privileges can. In addition, they can use the client to:

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

The NerveCenter Web Client

The following figure shows the GUI for the NerveCenter Web Client.

Figure 1-7. 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 NerveCenter Web Client to perform the tasks listed above and others, see the book *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.

Role in Network Management Strategy

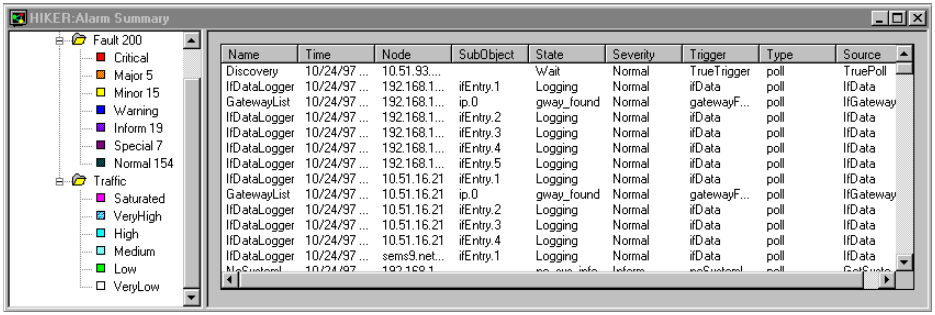
NerveCenter can play a variety of roles in an overall network management strategy. The role that NerveCenter plays in your strategy depends largely on the size of your network and on what other products you are using to manage your network and systems:

- 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 in a window information about active alarms. See the section *Standalone Operation* on page 22 for further information.
- For larger networks, multiple NerveCenters can be used in concert. For example, let's say that a company has a central site and three remote sites. Local NerveCenter systems could be set up to manage the 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 23 for further information.
- NerveCenter can be used in conjunction with a network management platform such as Hewlett Packard OpenView Network Node Manager, Hewlett Packard OpenView IT/Operations, CA Unicenter TNG, Tivoli TME, and Micromuse Netcool/OMNIbus which manages 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 24 for further information.
- NerveCenter is also tightly integrated with Hewlett Packard's OpenView Network Node Manager. In this situation, NerveCenter is responsible for SNMP trap handling, all polling activity, event correlation, and automated responses to conditions. See the section *Integration with NMPs for Node Information* on page 25 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 1-8.

Figure 1-8. NerveCenter's Alarm Console



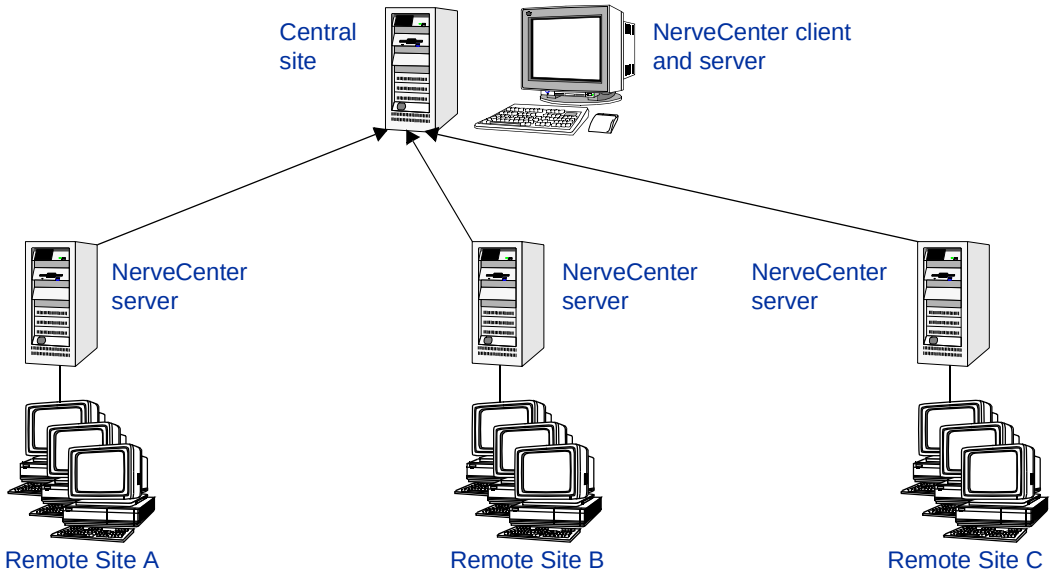
This console displays information about every current alarm instance. In addition, if you double-click on a line in the event console, you are taken to an Alarm History window that displays information about all of the alarm transitions that have occurred for the alarm instance you selected.

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 one 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 of the noteworthy network conditions at those remote sites.

Figure 1-9. Distributed NerveCenter Servers



This is a reliable solution because the remote NerveCenter servers use TCP/IP to notify the centrally located NerveCenter server of network conditions and retransmit messages as necessary to ensure their delivery.

There are a couple of 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.
- The remote NerveCenter servers can be run in lights-out mode. Being able to run NerveCenter lights-out 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 these messages, it can automatically take corrective action—such as broadcasting a command to a set of systems—or an operator can initiate this response.

You can integrate NerveCenter with the following network management platforms:

- CA Unicenter TNG
- Hewlett Packard OpenView IT/Operations
- Hewlett Packard OpenView Network Node Manager
- Micromuse Netcool/OMNIbus
- Tivoli Systems TME

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. See the later section, *Integration with NMPs for Node Information* on page 25, for more information.

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

Integration with NMPs for Node Information

If you're working at a larger site and need a topology map and more event history than NerveCenter provides, you can use NerveCenter with Hewlett Packard's OpenView Network Node Manager.

When used with OpenView Network Node Manager, NerveCenter can take its node information from the management platform and can be configured to take over all polling activity and event processing. NerveCenter's main task is to minimize the number of events that appear in the platform's event browser. NerveCenter does this by:

- Filtering out unimportant events
- Correlating related events and notifying the platform only of the underlying problem
- Handling problems through automated actions so that no notification is necessary

Figure 1-10 below shows an OpenView event browser that contains a flurry of events all caused by the same problem. Figure 1-11 shows what might appear in the browser if NerveCenter were used to screen and correlate the conditions and pass on only important information to the platform event browser.

Figure 1-10. Too Many Events

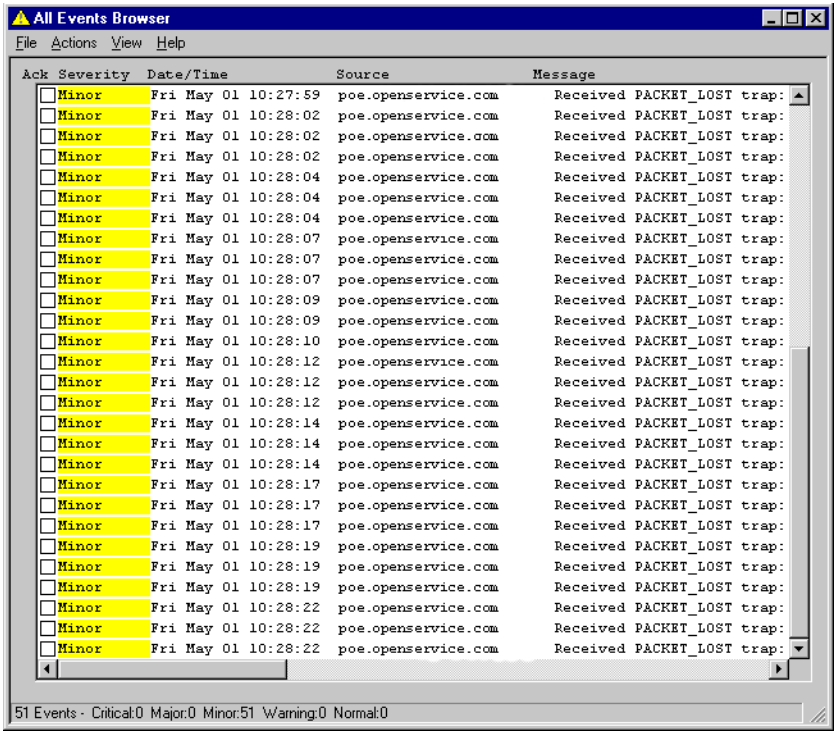
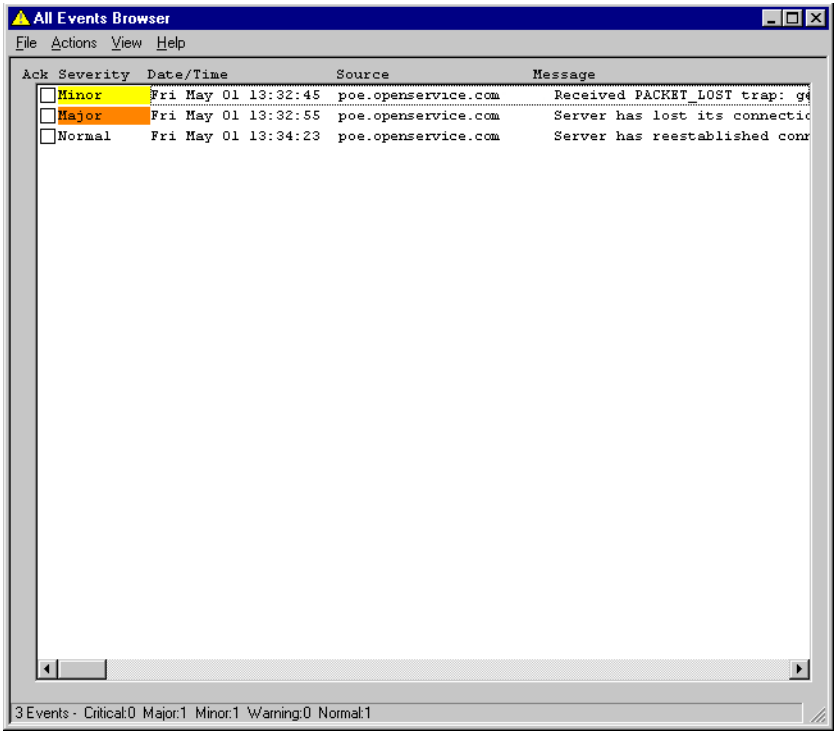


Figure 1-11. The Important Events



NerveCenter can also set the colors of nodes in the network management platform's map based on the severity of NerveCenter alarm states.

Administering NerveCenter 2

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 28	Explains the difference between basic users, power users, and administrators of NerveCenter.
NerveCenter Login Rights on page 28	Explains the difference between login rights as a user and as an administrator.
The Role of a NerveCenter Administrator on page 30	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:

- **NerveCenter basic users** – monitor NerveCenter using either NerveCenter Client or the Web Client. They have user login rights and should read the book *Monitoring Your Network*.
- **NerveCenter power users** – design and manage behavior models using the NerveCenter Client. They have administrator login rights and should read the books *Designing and Managing Behavior Models* and *Learning How to Create Behavior Models*.
- **NerveCenter 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 2-1 illustrates the differences between the three.

Table 2-1. 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 different 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 2-2 illustrates the difference between NerveCenter administrators and NerveCenter users login rights.

Table 2-2. 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 162.
- 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 148.
- 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 Information about the Status of a NerveCenter Server* on page 42. 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?
  - OpC Host: Is the connection to the machine hosting OpenView IT/Operations still available.
- 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 129.
- Keeping the NerveCenter serial numbers up to date. See *Working with NerveCenter Serial Numbers* on page 53.
- 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 Open Software Technical Support Web for news ideas and procedures at www.open.com.

Running the NerveCenter Server 3

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 32	Explains how to start and stop the NerveCenter Server on a UNIX platform.
Running the NerveCenter Server on Windows on page 32	Explains how to start and stop the NerveCenter Server on a Windows platform.
Troubleshooting: Running the NerveCenter Server on page 33	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.

❖ To start NerveCenter Server manually 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, where *install_path* is typically `/opt/OSInc`, type 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`**

For other command line switches see *Controlling NerveCenter from the Command Line* on page 229.

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.

❖ 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 chose Stop.
- From the command line, type:

  `ncstop -s`

  to stop the NerveCenter Service

  or

  `ncstop -All`

  which stops all running NerveCenter applications.

Troubleshooting: Running the NerveCenter Server

This section contains some common problems users 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.`**

See *Running the NerveCenter Server on UNIX* on page 32.

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 &.`**

See *Running the NerveCenter Server on UNIX* on page 32 or *Running the NerveCenter Server on Windows* on page 32.

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`**.

See *Running the NerveCenter Server on Windows* on page 32.

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`**.

For more infomration, see *ncstart* on page 253.

Connecting to the NerveCenter Server 4

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 36	Explains how to start the NerveCenter Administrator application.
Connecting Administrator to a NerveCenter Server on page 38	Explains how to connect a NerveCenter Administrator to a NerveCenter Server.
Connecting Administrator to More Than One NerveCenter Server on page 40	Explains the issues involved in connecting a single NerveCenter Administrator to more than one NerveCenter Server.
Deleting a Name from the Administrator's Server List on page 41	Explains how to delete a misspelled or old server name from the Administrator's server list.
Viewing Information about the Status of a NerveCenter Server on page 42	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 43	Explains how to disconnect the NerveCenter Administrator from a NerveCenter Server without closing the Administrator window.
Troubleshooting: Connecting to the NerveCenter Server on page 44	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 36
- *Starting the NerveCenter Administrator on Windows* on page 37

Starting the NerveCenter Administrator on UNIX

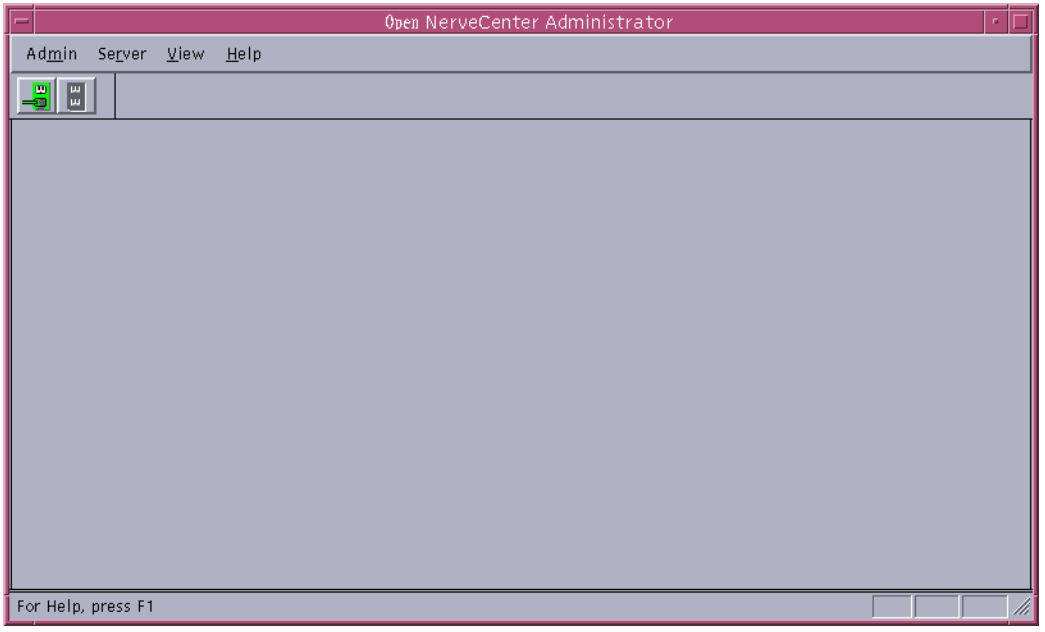
❖ To start the NerveCenter Administrator on UNIX:

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 32.

   A NerveCenter Administrator window opens.



Starting the NerveCenter Administrator on Windows

❖ To start the NerveCenter Administrator on Windows:

1. Select Start.

Note The following is the default path suggested during NerveCenter installation. It could be different.

2. Select Programs, then OpenService NerveCenter.
3. Select Administrator.

   A NerveCenter Administrator window opens.

Tip It is also possible to start the NerveCenter Administrator from a command prompt, 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.

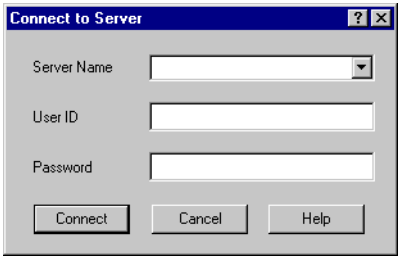
You can connect your NerveCenter Administrator to more than one server at one time. However, only one NerveCenter Server can be the *active server* at a time. The active server determines which NerveCenter database is used when you are configuring settings.

❖ To connect a NerveCenter Administrator to a NerveCenter Server:

1. Open NerveCenter Administrator. See *Starting NerveCenter Administrator* on page 36.

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's Server List* on page 41.)

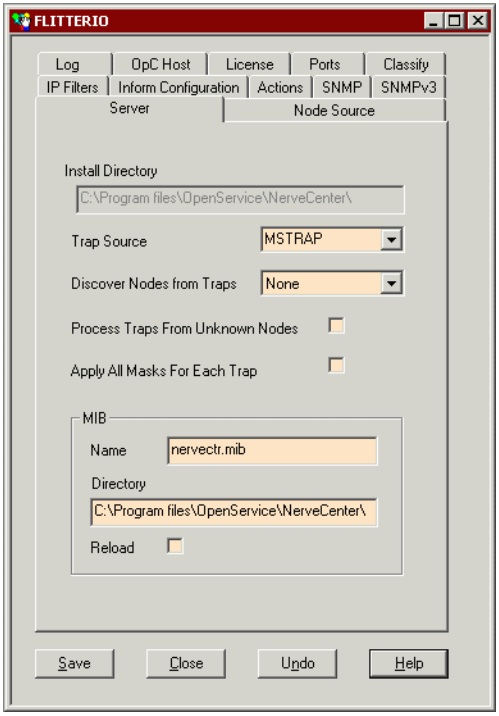
4. In the User ID and Password fields, type an appropriate user name and password or leave these fields blank.

Tip If you are running NerveCenter Administrator on a Windows machine and you want to connect to a NerveCenter Server using the same user name and password you used to log in to Windows, you can leave these fields blank. Otherwise, you must enter a user name and password.

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 31 for more details.

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

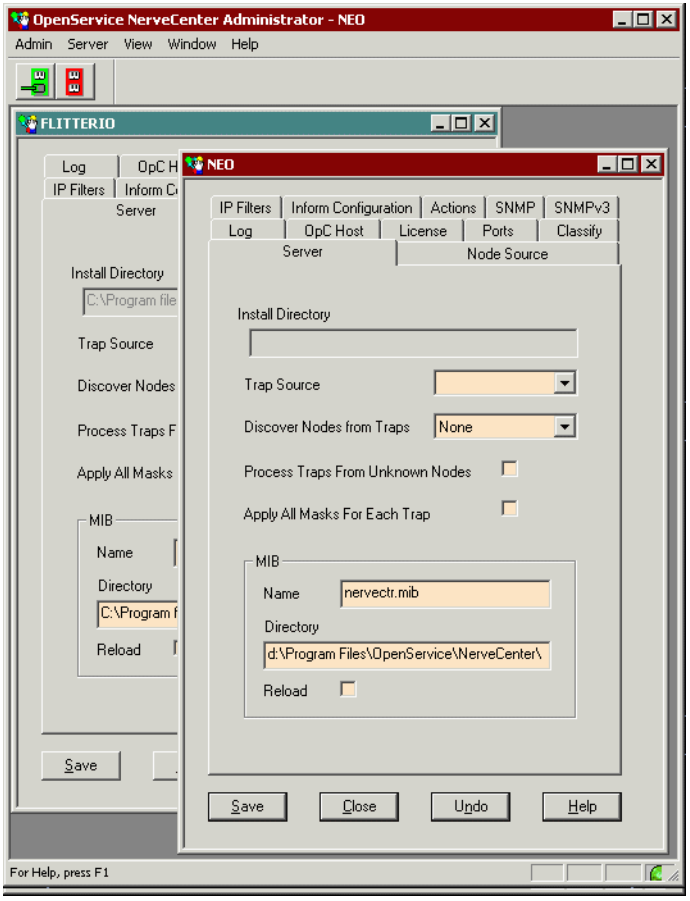


Connecting Administrator to More Than One NerveCenter Server

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, follow the steps in the section *Connecting Administrator to a NerveCenter Server* on page 38 for each NerveCenter Server. NerveCenter Administrator will display a window for each connected NerveCenter Server. You will probably need to move the windows to view each Server separately.

Figure 4-1. NerveCenter Administrator Connected to More Than One 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's 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.

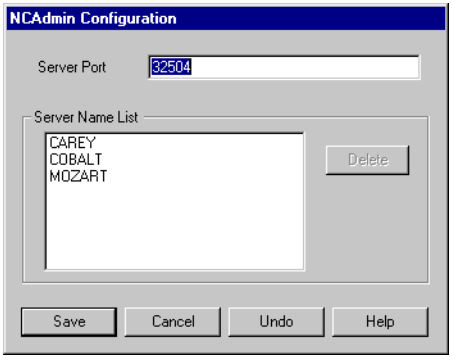
As helpful as this list is, there may be times when you need to delete a server from this list. It may be the name of a server that you will never connect to again. It could also be the misspelled name of a server you were unable to connect to because of the misspelling.

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

1. Open NerveCenter Administrator. See *Starting NerveCenter Administrator* on page 36.

2. From the Admin menu, select Configuration.

   The NCAdmin Configuration window is displayed.



   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 Information about the Status of a NerveCenter Server

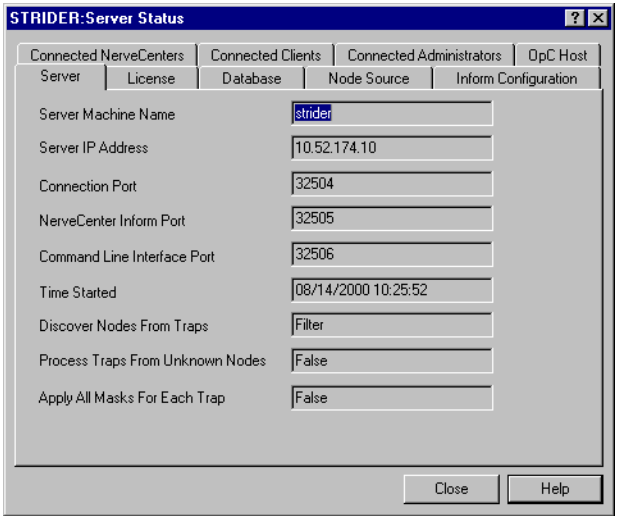
The Server Status window of NerveCenter Administrator and Client allow you to see current information pertaining to an active NerveCenter Server.

❖ 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 38.

2. From the Server menu, select Server Status.

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



3. To close the window, select Close.

The Server Status window has several tabs that display read-only data about the active NerveCenter Server. Table 4-1 describes the Server Status Tabs.

Table 4-1. Server Status Tabs

Server	Displays information about the connected NerveCenter Server, such as its ports and basic settings.
License	Displays information about the serial numbers associated with the active NerveCenter Server.
Database	Displays information about the database this NerveCenter Server is using. Includes statistics on its current behavior models.

Table 4-1. Server Status Tabs (continued)

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.
OpC Host	Displays information, if applicable, about the NerveCenter's HP OpenView IT/Operations host.

Disconnecting the Administrator from a NerveCenter Server

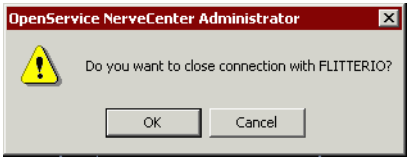
When you close a NerveCenter Administrator, all connections to NerveCenter servers are broken. However, you may also want to disconnect the administrator from a server without stopping a NerveCenter Administrator.

❖ 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.



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 32.

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 32 or *Running the NerveCenter Server on Windows* on page 32.

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's Server List* on page 41.

Managing the NerveCenter Server 5

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 61.

This chapter includes the following sections:

Section	Description
Managing the NerveCenter Trap Source on page 46	Explains how to determine and change the trap source, or trap engine, you want to use for capturing SNMP traps.
Working with NerveCenter Serial Numbers on page 53	Explains the purpose of NerveCenter serial numbers and how to add and delete them.
Reconfiguring a NerveCenter Server from the Command Line on page 57	Explains how to change settings to a NerveCenter Server by importing an .ini file.
Troubleshooting: Managing the NerveCenter Server on page 60	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 SNMP v3.

See the following sections for more information:

- *Trap Source Overview* on page 46
- *Changing the SNMP Trap Source* on page 47

Trap Source Overview

During installation, you indicated whether NerveCenter should be configured to support SNMP v3. 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 5-1. NerveCenter SNMP v3 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 SNMP v3 support in NerveCenter, 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 SNMP v3.

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

- OVTrapD: SNMP v1 and v2C traps
- MS Trap Service: SNMP v1 traps
- NerveCenter: SNMP v1, v2C, and v3 traps

To manage SNMP v3 traps, you must use NerveCenter as a trap source.

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.

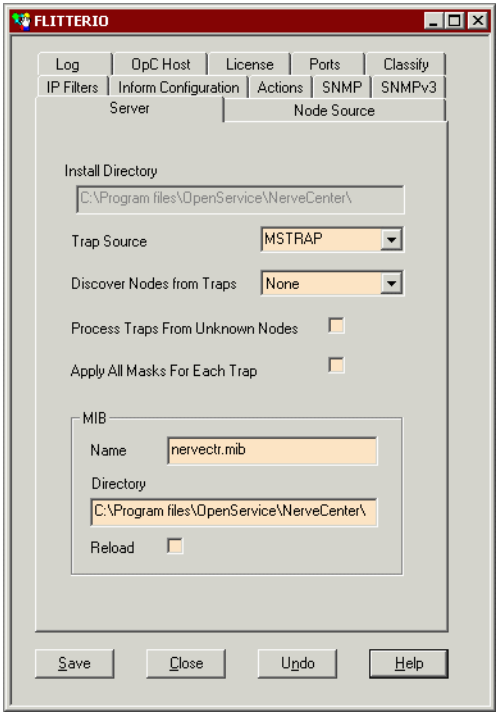
❖ To change the NerveCenter trap source:

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

   See *Connecting Administrator to a NerveCenter Server* on page 38.

2. Select the Server tab.

   The Server page is displayed.



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 the table below for details.

7. Restart the NerveCenter Server.

The Table 5-2 summarizes what to do when changing the trap source.

Table 5-2. 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 51, except 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 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 *Managing NerveCenter*.
   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 *Managing NerveCenter*.
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 51.
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 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:

Windows 2000 Professional

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.

Windows NT 4.0

1. Open the Service control panel by selection the following from the Start menu:

   Settings > Control Panel > Services

2. Select the SNMP Trap Service.

3. Select Startup.

4. Under Startup Type, select the Disabled radio button.

5. Select OK.

6. Select Close.

Working with NerveCenter Serial Numbers

Every NerveCenter Server needs at least one serial number. Each NerveCenter serial number provides:

- The number of NerveCenter Clients that are permitted to connect to that NerveCenter Server
- The number of nodes that the NerveCenter Server can monitor

You can purchase new serial numbers from your authorized sales reseller or representative to increase the number of NerveCenter Clients that can connect to a NerveCenter Server and/or the number of nodes that can be monitored by a NerveCenter Server.

The sequencing of the serial number keys is not guaranteed. There is no way to tell which single serial number is active at any instant. The total number of nodes and Clients that can connect to the Server are a cumulative figure of all the serial numbers that have been accepted.

When managing NerveCenter serial numbers, an administrator may be expected to perform two actions:

- *Adding a Serial Number to a NerveCenter Server* on page 54
- *Deleting an Old Serial Number from a NerveCenter Server* on page 55

To learn about the rules NerveCenter follows and restrictions imposed when adding and deleting serial numbers, see *NerveCenter Rules for Managing Serial Numbers* on page 56.

Adding a Serial Number to a NerveCenter Server

Once you obtain a new serial number from your authorized sales representative, you will need to add the serial number to the appropriate NerveCenter Server.

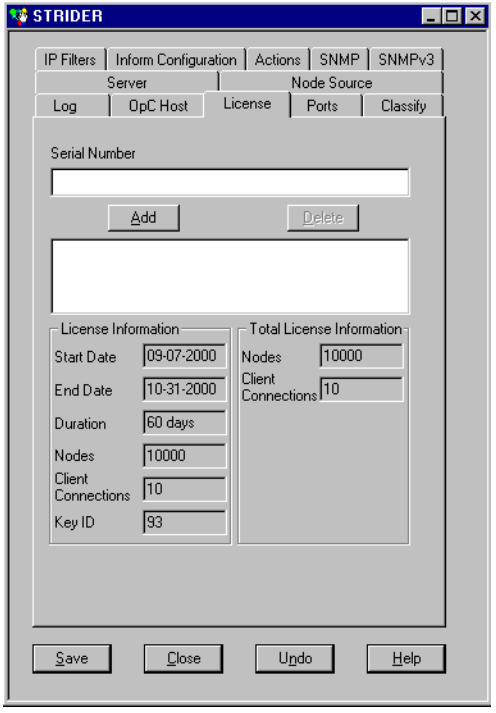
Note You can also add a NerveCenter serial number using the ImportUtil tool. See *Reconfiguring a NerveCenter Server from the Command Line* on page 57.

❖ To add a new serial number to a NerveCenter Server:

1. Open NerveCenter Administrator and connect to the appropriate Server. See *Connecting Administrator to a NerveCenter Server* on page 38.

2. Select the License tab.

   The License tab is displayed.



   The license tab of administrator allows you to view information about current serial numbers, delete old serial numbers, and add new serial numbers.

3. In the Serial Number field, type the new serial number.

4. Select Add.

   Licensing information associated with this serial number is displayed at the bottom of the License tab.

5. If you have more than one serial number, repeat steps 3 and 4.

6. Select Save.

The NerveCenter Server monitors as many nodes and establishes a connection with as many NerveCenter Clients as the serial number permits.

Deleting an Old Serial Number from a NerveCenter Server

As a NerveCenter serial number expires, you may want to remove it from the NerveCenter Server's data.

Note NerveCenter will not allow you to delete your last serial number. If you want to replace an evaluation serial number with a permanent one, add the permanent number first and then you can delete the evaluation number.

❖ To delete an old serial number from a NerveCenter Server:

1. Open NerveCenter Administrator and connect to the appropriate Server. See *Connecting Administrator to a NerveCenter Server* on page 38.

2. Select the License tab.

   The License tab is displayed.

3. From the list of serial numbers, highlight the serial number you want to remove.

4. Select Delete.

   The serial number is removed from the serial number list.

5. Select Save.

The serial number is removed from the NerveCenter Server.

NerveCenter Rules for Managing Serial Numbers

NerveCenter tracks how many serial numbers are registered for a particular NerveCenter Server and what types of serial numbers they are (permanent or evaluation). NerveCenter follows certain rules to manage serial numbers. For example, when removing serial numbers, NerveCenter will not allow you to delete your last serial number.

The following table summarizes the rules that NerveCenter follows to manage serial numbers.

Table 5-3. NerveCenter Rules for Serial Numbers

If you currently have ...	NerveCenter does the following ...
No serial number	◆ Accepts either an evaluation or permanent serial number. ◆ Does not manage a network until you have added at least one valid evaluation or permanent serial number.
Evaluation serial number	◆ Accepts a permanent serial number to replace the evaluation number. Add the permanent number first and then you can delete the evaluation number. ◆ Does not accept any additional evaluation serial numbers. ◆ Does not allow you to delete the last serial number. ◆ Does not accept duplicate entries for the same serial number.
Expired evaluation serial number	◆ Automatically removes the expired number from the list and reduces accordingly the total number nodes and clients that are managed. ◆ Accepts either a valid evaluation or permanent serial number.
One or more permanent serial numbers	◆ Accepts any number of additional permanent serial numbers. ◆ Accepts a single valid evaluation serial number; a second evaluation serial number will not be accepted. ◆ Allows you to delete any of the serial numbers except the last one. ◆ Does not accept duplicate entries for the same serial number.

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. 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.

❖ 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 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.

   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 either the section **[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.

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
   ```

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
   ```

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 5-4.

Table 5-4. Information About imputil.ini Sections

For Information About...	Refer to This Book...
CONFIG_PLATFORM	*Integrating NerveCenter with a Network Management Platform*
CONFIG_PLATFORM_MAPSUBNETS	*Integrating NerveCenter with a Network Management Platform*
CONFIG_PLATFORM_NETNODENOTIFY	*Integrating NerveCenter with a Network Management Platform*
CONFIG_SERVER	*Managing NerveCenter*
CONFIG_SERVER_PATH	*Managing NerveCenter*
CONFIG_SERVER_PAGER	*Managing NerveCenter*
CONFIG_SERVER_SNMP	*Managing NerveCenter*

Table 5-4. Information About imputil.ini Sections (continued)

For Information About...	Refer to This Book...
CONFIG_SERVER_SNMPV3	*Managing NerveCenter*
CONFIG_SERVER_LOGS	*Managing NerveCenter*
CONFIG_SERVER_MSMAIL	*Managing NerveCenter*
CONFIG_SERVER_SMTPMAIL	*Managing NerveCenter*
IMPORT_MODEL	*Designing and Managing Behavior Models*
IMPORT_NODE	*Designing and Managing Behavior Models*

Troubleshooting: Managing the NerveCenter Server

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

When I try to connect to a NerveCenter Server it tells me: Number of allowed client connection exceeded

Problem: The NerveCenter Server's license has expired or is inadequate for your needs.

Solution: Contact your authorized NerveCenter reseller or representative to obtain a new serial number.

See *Working with NerveCenter Serial Numbers* on page 53.

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 57.

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 57.

Managing NerveCenter Port Settings 6

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
NerveCenter's Client/Server Communication Port on page 62	Explains how to change the port at which NerveCenter Server communicates with its client applications.
Configuring NerveCenter to Receive Inform Actions on page 66	Explains how to configure a NerveCenter Server to receive an inform action from another NerveCenter Server.
Changing the Command Line Interface Port on page 68	Explains how to change the port NerveCenter uses for its command line interface.
Specifying SNMP Ports for NerveCenter on page 70	Describes how to configure the ports NerveCenter uses to receive SNMP traps and send/receive SNMP correspondences.
Troubleshooting: Managing NerveCenter Port Connections on page 72	Provides troubleshooting tips related to ports.

NerveCenter's Client/Server Communication Port

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 62
- *Changing the NerveCenter Administrator's Server Port* on page 64
- *Changing the NerveCenter Client's Server Port* on page 65

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.

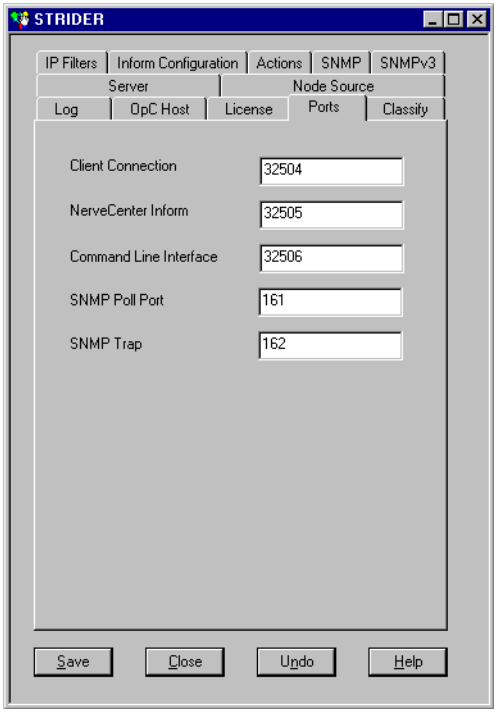
❖ To change the NerveCenter Server connection port:

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

   See *Connecting Administrator to a NerveCenter Server* on page 38.

2. Select the Ports tab.

The Ports tab is displayed.



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

Caution Remember the port number. If you change it here, you must make corresponding 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 31.

NerveCenter uses the specified port when connecting with any NerveCenter Administrator or Client.

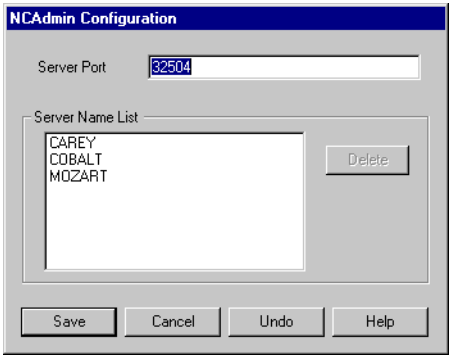
Changing the NerveCenter Administrator's Server Port

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 36.
2. From the Admin menu, choose Configuration.

   The NCAdmin Configuration window appears.



   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's Server Port

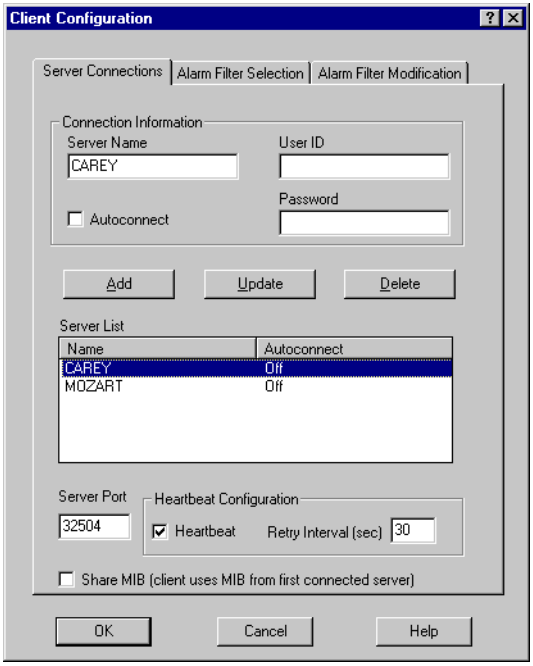
If you change a NerveCenter Server's communication port number, be sure all applications, such as NerveCenter Client, have a matching port number.

❖ To change the NerveCenter Client's server port:

1. Open NerveCenter Client. See *Designing and Managing Behavior Models* in the NerveCenter documentation for more details.

2. In the Client menu, choose Configuration.

   The Client Configuration window appears.



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.

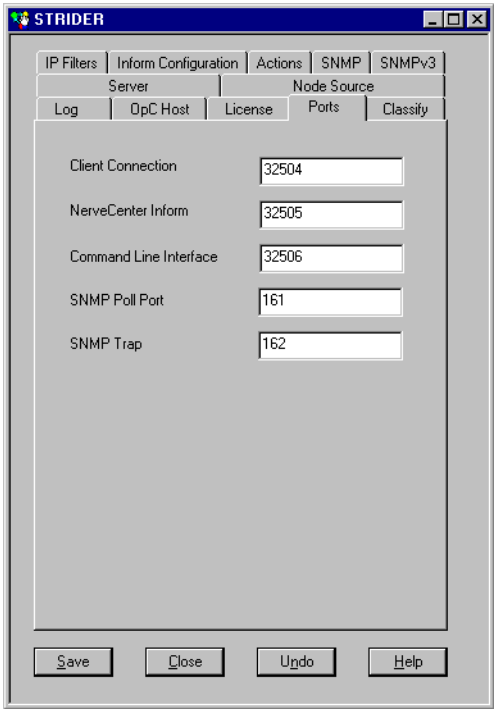
Before a NerveCenter Server can receive a trap it must be configured to have a listening port number. By default, this port number is 32505.

❖ 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 38.

2. Select the Ports tab.

The Ports tab is displayed.



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 31.

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

❖ **To view all the NerveCenter Servers configured to send an Inform to the current server:**

1. In either NerveCenter Administrator or NerveCenter Client, from the Server menu, 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

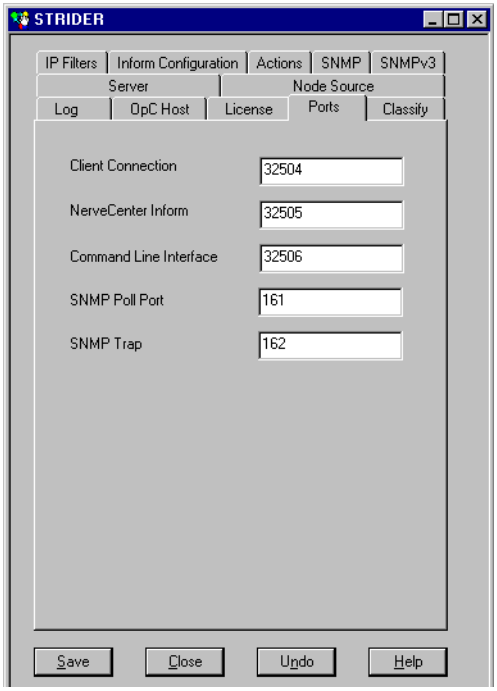
You can use NerveCenter's command line interface to delete, list, or set (enable or disable) alarms, trap masks, nodes, and polls from a Windows Command Prompt or a UNIX shell. You can also connect to, display the status of, and disconnect from NerveCenter servers using the CLI. You can issue commands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

❖ **To specify the port NerveCenter uses for the command line interface:**

1. Open NerveCenter Administrator and connect to an appropriate NerveCenter Server. See *Connecting Administrator to a NerveCenter Server* on page 38.

2. Select the Ports tab.

The Ports tab is displayed.



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.

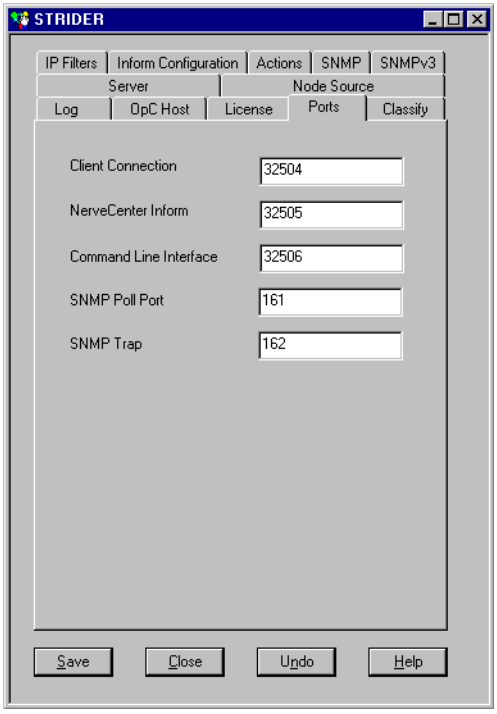
You can change the ports that NerveCenter uses for receiving traps and sending polls.

❖ To specify 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 38.

2. Select the Ports tab.

   The Ports tab is displayed.



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.

Open 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 SNMP v3 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 32 or *Running the NerveCenter Server on Windows* on page 32.

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 66.

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 32 or *Running the NerveCenter Server on Windows* on page 32.

Managing Node Data 7

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
NerveCenter's Node List on page 74	Explains that NerveCenter stores basic information about each node it will monitor in a node list in the NerveCenter database.
Filtering Nodes on page 75	Explains how to specify which managed nodes NerveCenter is responsible for monitoring.
Processing Traps from Unknown Nodes on page 83	Explains how to configure NerveCenter to receive and process traps from nodes not included in its node list.
Populating the Node List Initially on page 85	Explains the methods used to populate NerveCenter's node list with initial node data.
Maintaining the Node List on page 89	Explains the various ways nodes can be added to and deleted from NerveCenter's database.
Troubleshooting: Managing node data on page 95	Lists common problems NerveCenter users face when managing node data.

NerveCenter's 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 SNMP v1, v2c, and v3, when NerveCenter obtains its nodes from a platform, the platform does not provide SNMP version information. By default, NerveCenter deems the SNMP agents on these nodes to be SNMP v1 by default.

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 *NerveCenter: Node Classification* for more information about SNMP auto-classification.

Note If you want NerveCenter to manage SNMP v3 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 47 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 several 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. NerveCenter does not need to monitor every node on your network. Instead you, as the administrator, can specify which managed nodes NerveCenter will be responsible for monitoring.

There are several methods for restricting which nodes will be placed in NerveCenter's node list:

- *Filtering Using a Node's Capabilities* on page 75
- *Filtering Using a Node's System Object Identifier* on page 77
- *Filtering nodes using a Node's IP Address* on page 78

Filtering Using a Node's Capabilities

NerveCenter allows you to monitor managed nodes that have particular capabilities. Typically your network management platform assigns these capabilities to a node to determine applicable management activities. Some examples of these capabilities are isRouter, isHub, and isIP.

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

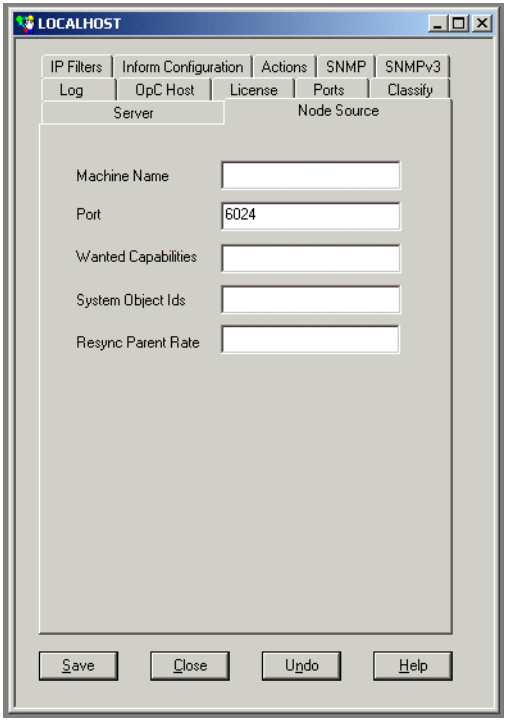
❖ To filter using a node's capabilities:

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

   See the section *Connecting Administrator to a NerveCenter Server* on page 38.

2. Select the Node Source tab.

The Node Source tab is displayed.



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 performs a resynchronization with your network management platform's database.

New nodes will be added. Any node that is marked Autodelete (the default) will be deleted.

Filtering Using a Node's 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.

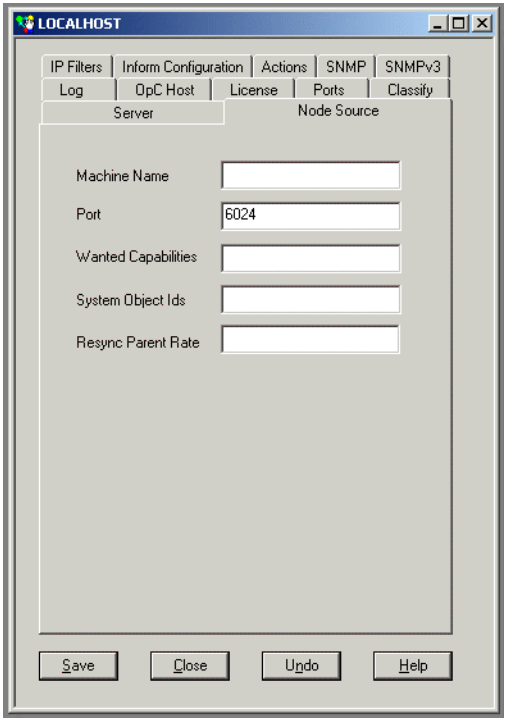
❖ To filter using a node's system object identifier:

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

   See the section *Connecting Administrator to a NerveCenter Server* on page 38.

2. Select the Node Source tab.

   NerveCenter displays 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 using a Node's 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 7-1 and Table 7-2 illustrate some filter configurations and their results:

Table 7-1. 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 7-2. 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 7-3 illustrates some filter configurations with exclusions and their results.:

Table 7-3. 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 determine automatically or manually subnet criteria used to filter nodes by IP address.

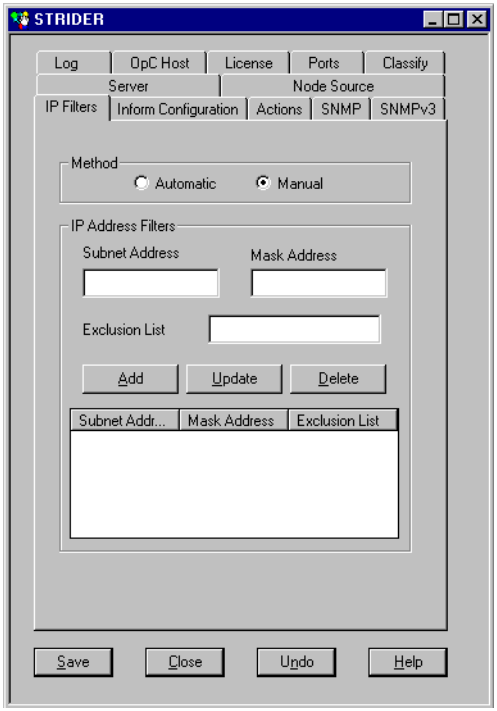
❖ To configure NerveCenter to determine subnet criteria automatically:

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

   See the section *Connecting Administrator to a NerveCenter Server* on page 38.

2. Select the IP Filters tab.

   The IP Filters tab is displayed.



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.

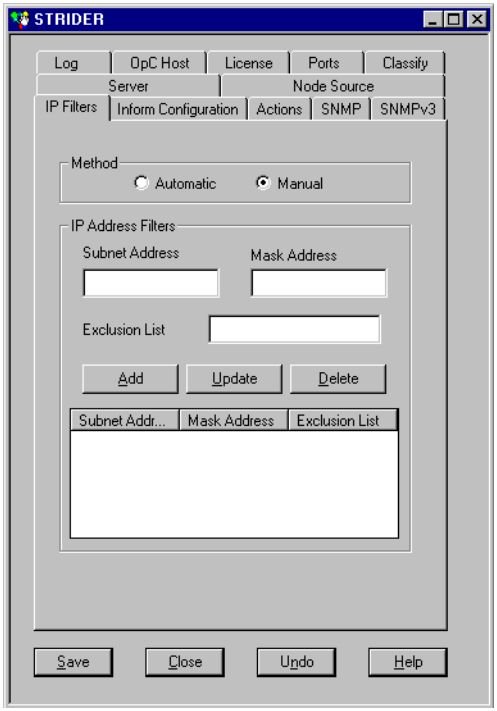
❖ To set the subnet criteria manually:

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

   See the section *Connecting Administrator to a NerveCenter Server* on page 38.

2. Select the IP Filters tab.

   The IP Filters tab is displayed.



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.

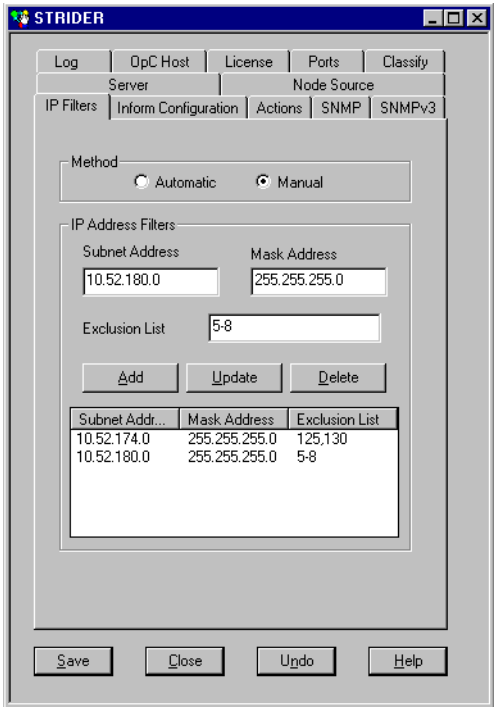
   A node's subnet address combines the node's IP address with the subnet mask.

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.



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.

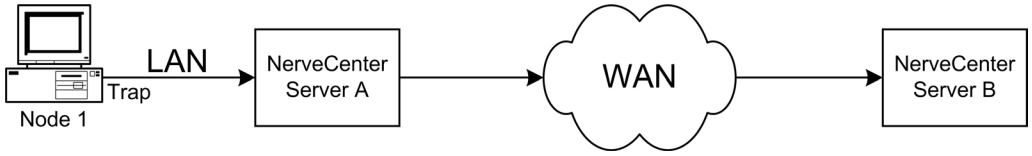
Processing Traps from Unknown Nodes

NerveCenter offers a lot of flexibility in filtering the managed nodes it monitors. See *Filtering Nodes* on page 75. By setting these 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 7-1. 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?

NerveCenter's 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 If you want to learn how a NerveCenter Server can add an unknown node to its node list when it receives a trap from that node, see the section *Adding Nodes Discovered from Traps* on page 91.

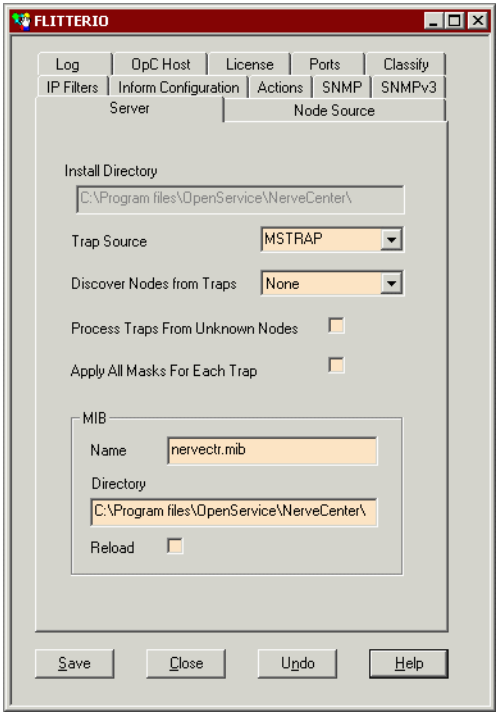
❖ To process traps from unknown nodes:

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

   See *Connecting Administrator to a NerveCenter Server* on page 38.

2. Select the Server tab.

   NerveCenter displays the Server tab.



3. Select the Process Traps From Unknown Nodes box.

   If this box is not selected, then NerveCenter will ignore any trap it receives from a node not included 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.

Populating the Node List Initially

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 75), you must tell NerveCenter where to look for the initial data it will use to populate its node list.

There are several methods for populating NerveCenter's node list:

- *Populating NerveCenter's node list using your network management platform as a node data source* on page 85
- *Populating Using the IPSweep Behavior Model* on page 87
- *Populating the Node List Manually* on page 89

Populating NerveCenter's node list using your network management platform as a node 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 *Designing and Managing Behavior Models*.) After NerveCenter initially populates its node list, any subsequent mapping of OIDs to property groups will affect only new nodes added to the node list.

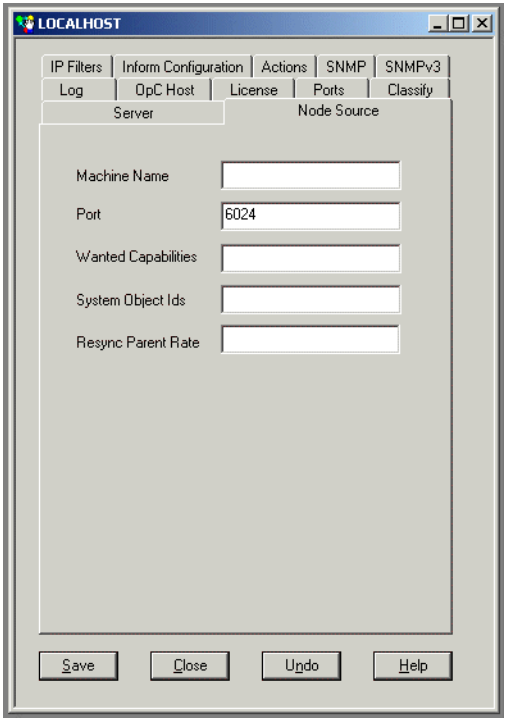
❖ To use your network management platform as a node data source:

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

   See the section *Connecting Administrator to a NerveCenter Server* on page 38.

2. Select the Node Source tab.

NerveCenter displays the Node Source tab.



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

   For example, if you have your network management platform running on a machine named Norm, you would type the name **`Norm`** or its IP address in the Machine Name field.

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

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.

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.

For situations such as these, NerveCenter provides the IPSweep behavior model. Using the IPSweep behavior model, NerveCenter detects unknown nodes and adds them to the node database, provided they fall within the set filters.

Note If you are using NerveCenter in Windows configured with Domain Name Server (DNS), the IPSweep behavior model requires that Enable DNS for Windows Resolution be selected. (This option is found on the WINS address tabbed page of the TCP/IP protocol, found on the Protocols tabbed page in the Network applet in the Control Panel.)

❖ To populate the node list using the IPSweep behavior model:

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

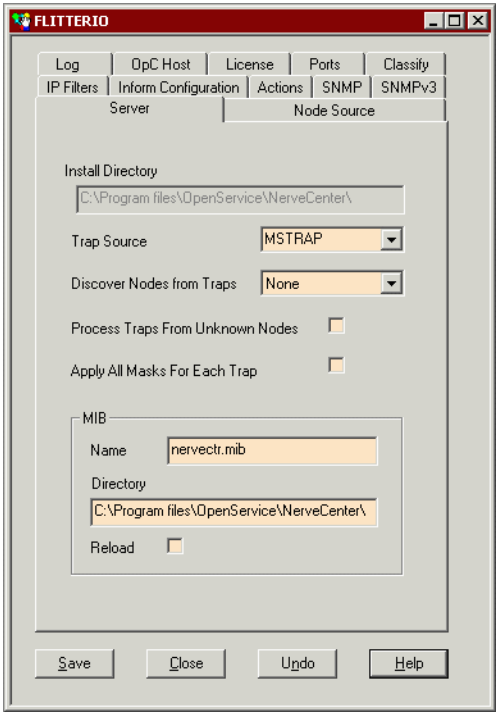
   See *Connecting Administrator to a NerveCenter Server* on page 38.

2. Select the IP Filters tab and configure NerveCenter to filter by the appropriate subnet criteria. See *Filtering nodes using a Node's IP Address* on page 78.

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 Filter.

Note If the Discover Nodes from Traps field is set to None, the IPSweep behavior model will not work.

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

6. From NerveCenter Client, turn on the IPSweep behavior model. See the *Designing and Managing Behavior Models* in the NerveCenter documentation 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.

Populating the Node List Manually

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

- *Populating NerveCenter's node list using your network management platform as a node data source* on page 85
- *Populating Using the IPSweep Behavior Model* on page 87

An alternative is to populate the NerveCenter node list by hand. For each node you want NerveCenter to 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 the section *Adding and Deleting Nodes Manually* on page 93 for each node you want to include in NerveCenter's node list.

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 *Populating the Node List Initially* on page 85), 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 90
- *Adding Nodes Discovered from Traps* on page 91
- *Adding and Deleting Nodes Manually* on page 93

The following chart illustrates how each method affects NerveCenter's node list.

Table 7-4. 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 the section "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 NerveCenter's node list using your network management platform as a node data source* on page 85 for more details on how to declare a node data source.)

The Node Source and IP Filters tabs also specify the parameters NerveCenter uses to filter node data. (See *Filtering Nodes* on page 75.)

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's maps but is managed in another, the node will remain 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. (See the section *Processing Traps from Unknown Nodes* on page 83)
- Whether to add the unknown node to its node list. This section explains how to configure this setting.

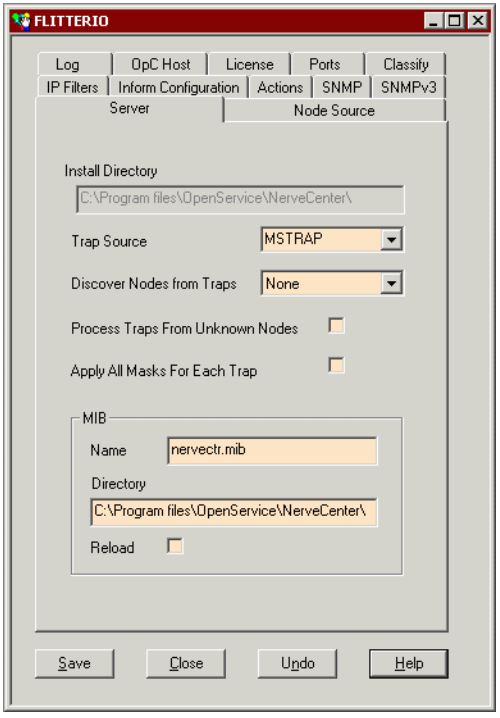
❖ To configure the way unknown nodes affect NerveCenter's node list:

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

   See *Connecting Administrator to a NerveCenter Server* on page 38.

2. Select the Server tab.

   NerveCenter displays the Server tab.



3. In the Discover Nodes from Traps list, select which nodes NerveCenter should place in its node list:

   - With a None setting, NerveCenter will not add any unknown nodes to the node list.
   - With a Filter setting, NerveCenter will add to its node list an unknown node sending a trap *only* if it meets the criteria established by the node filters.
   - With an All setting, NerveCenter adds to its node list any unknown node sending a trap, regardless of the node filters.

4. 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 to autodelete 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:

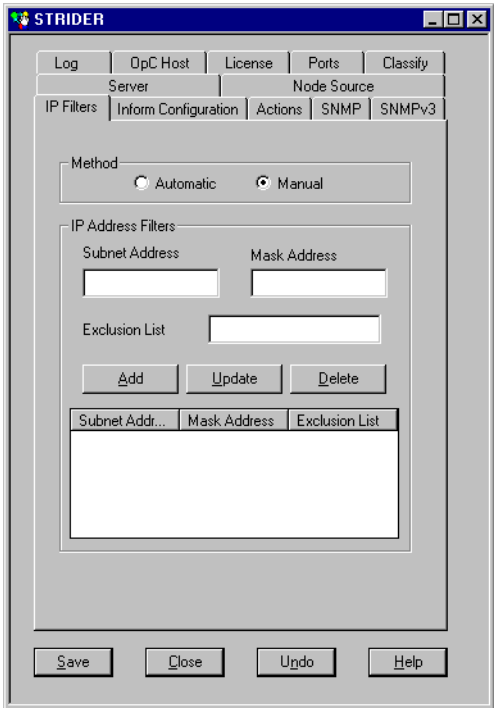
- Adding a Node Manually—You add nodes in the NerveCenter Client module. To learn how to add a node manually, refer to the instructions in *Designing and Managing Behavior Models*.
- Deleting a Node Manually—You delete nodes in the NerveCenter Client module. To learn how to delete a node manually, refer to the instructions 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. See *Connecting Administrator to a NerveCenter Server* on page 38.

2. From the Admin menu, choose IP Filters.

   The IP Filters tab is displayed.



3. From the IP Address Filters list, select the subnet address that contains the node or nodes you wish to be permanently deleted.

4. In the Exclusion List field, enter all the nodes you want permanently deleted from the node list. 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.

Note 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.

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 the book *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 93 or *Synchronization with your network management platform* on page 90.

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 91.

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 87.

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 NerveCenter's node list using your network management platform as a node data source* on page 85.

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 87.

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 using a Node's IP Address* on page 78.

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 91.

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 90.

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 93.

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 using a Node's IP Address* on page 78.

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 *Designing and Managing Behavior Models* for more details.

Managing SNMP Settings 8

This section contains information you need to configure NerveCenter for SNMP communication and to support SNMP v3 agents.

This chapter includes the following sections:

Section	Description
Specifying SNMP Poll Intervals for NerveCenter on page 100	Explains how to manage SNMP performance by controlling polling intervals.
Specifying SNMP Ports for NerveCenter on page 101	Describes how to configure the ports NerveCenter uses to receive SNMP traps and send/receive SNMP correspondences.
Overview of NerveCenter SNMP v3 Support on page 103	Summarizes NerveCenter support for SNMP v3 and points to where you can find information about specific settings and requirements.
Discovery and Initialization of SNMP v3 Agents on page 107	Summarizes the requirements for discovery and initialization of SNMP v3 agents.
Configuring an SNMP v3 Agent for NerveCenter on page 108	Summarizes the requirements for configuring an SNMP v3 agent for polling.
Configuring an Initial User for Discovering an SNMP v3 Agent on page 110	Summarizes how you configure an initial user without security restrictions so an SNMP v3 agent can be discovered.
SNMP Auto and Manual Classification Settings on page 110	Describes node classification and explains how to enable auto-classification of SNMP versions.
Setting the Maximum SNMP v3 Requests per Cycle on page 115	Explains how to manage SNMP v3 performance by controlling the maximum number of requests per cycle.
SNMP v3 Security Settings on page 117	Describes how to change the authentication and privacy key passwords or the NerveCenter user and context.
SNMP v3 Operations Log on page 121	Describes the Operations Log that records SNMP v3 operations and errors that occur while attempting to perform those operations.
SNMP Error Status on page 125	Describes SNMP v3 error status messages and indicates which ones cause polling to stop for a node.

Specifying SNMP Poll Intervals for NerveCenter

NerveCenter sends polls to SNMP agents to obtain MIB information. You can change settings that affect how often NerveCenter reissues a poll and at what interval. By default the number of retry attempts is three. For example, let us say the number of retries were set to three and a poll was attempted of a device that could not respond. The NerveCenter Server would make four polling attempts, the first initial followed by three retries.

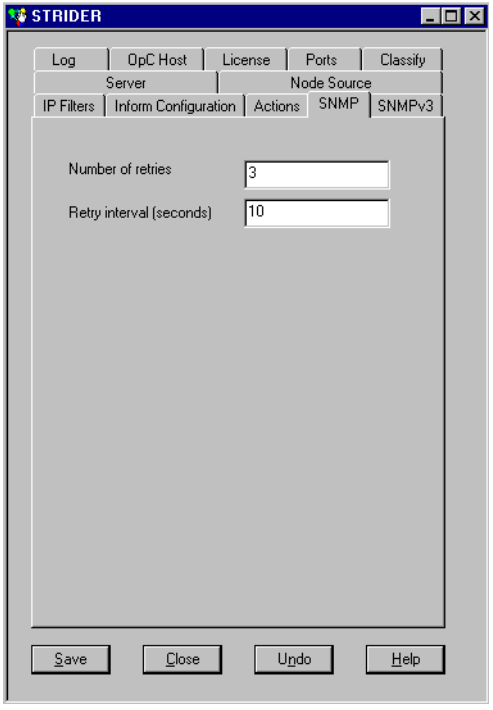
The retry interval is significant in that the polling rate should be high enough to account for the number of retries at each interval. For example, if your NerveCenter Server is set to retry three times at a 30 second interval, the polling rate for any of its behavior model's polls should be no lower than a minute and a half (three retry intervals multiplied by 30 seconds).

❖ To specify poll settings for NerveCenter:

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server. See *Connecting Administrator to a NerveCenter Server* on page 38.

2. Select the SNMP tab.

   The SNMP tab is displayed.



3. In the Number of retries field, type the number of times you want NerveCenter to reissue an unanswered SNMP or ICMP request poll.

4. In the Retry interval field, type 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.

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.

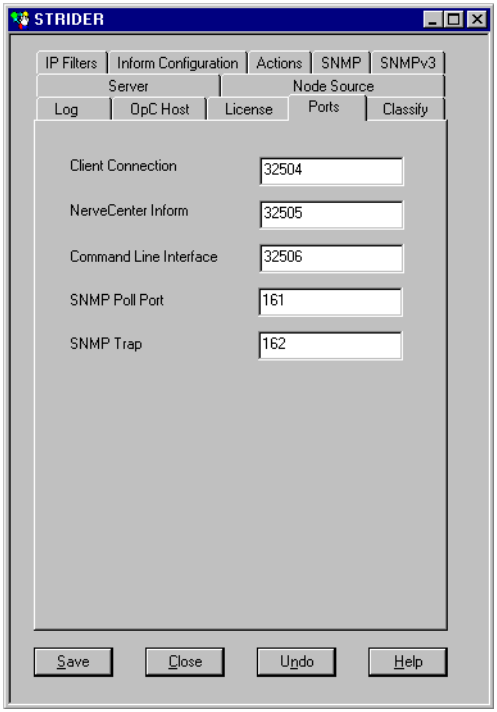
You can change the ports that NerveCenter uses for receiving traps and sending polls.

❖ To specify 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 38.

2. Select the Ports tab.

The Ports tab is displayed.



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 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 SNMP v3 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 will handle SNMP data according to these settings.

Overview of NerveCenter SNMP v3 Support

NerveCenter support for SNMP v2c (community-based SNMP v2) and v3 includes new data types and enhanced security for communication. SNMP v1 and v2c rely on community names for authentication. SNMP v3 enhances authentication and expands its services to include privacy. SNMP v3 expands on the earlier concept of MIB views to control access to management information. SNMP v3 uses a View-based Access Control Model (VACM) to determine the level of access a user has for viewing MIB data.

Following are highlights of NerveCenter support for SNMP v2c/v3:

- Before NerveCenter can discover SNMP v3 agents on nodes, the nodes must have an initial user configured for discovery.

  See *Configuring an Initial User for Discovering an SNMP v3 Agent* on page 110.

  Refer to the book *Designing and Managing Behavior Models* for details about testing communication with a node using the NerveCenter Test Version poll.

- NerveCenter communicates (sends polls) with an SNMP v3 agent on behalf of a specified NerveCenter user in a defined context. Before NerveCenter can poll SNMP v3 agents, the agents must be configured to support the NerveCenter user and context. By default, the user name is NCUser and the context is NCContext, though you can change both in NerveCenter.

  See *Configuring an SNMP v3 Agent for NerveCenter* on page 108.

  See *Changing the NerveCenter SNMP v3 User Name and Context* on page 119.

- NerveCenter supports three security levels for communicating with SNMP v3 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 SNMP v3 Security* on page 105 for details about security.

  Refer to the book *Designing and Managing Behavior Models* for details about setting a node's security level.

- The authentication and privacy protocols require specialized keys, called authentication and privacy keys. These keys are generated from corresponding passwords. You can change these passwords in NerveCenter, thereby changing the keys. When changing keys in NerveCenter, you can command NerveCenter to update the key changes on all nodes.

  See *NerveCenter Support for SNMP v3 Digest Keys and Passwords* on page 106.

  See *Changing the SNMP v3 Key Passwords* on page 117.

- NerveCenter supports either HMAC-MD5-96 (MD5) or HMAC-SHA-96 (SHA) as authentication protocol on a per-node basis and CBC-DES as the privacy protocol. The default authentication protocol for NerveCenter is MD5. If you change the authentication protocol on an SNMP v3 agent, you must likewise change the protocol used by NerveCenter to manage the corresponding node in its database.

  Refer to the book *Designing and Managing Behavior Models* for details about changing the authentication protocol used by NerveCenter for an agent.

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

Note Auto-classification is disabled when you install NerveCenter. You must enable this feature before NerveCenter can classify nodes added to its database.

See *SNMP Auto and Manual Classification Settings* on page 110.

Refer to the book *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 46.

- SNMP v3 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 *SNMP v3 Operations Log* on page 121.

  See *SNMP Error Status* on page 125 for information about SNMP v3 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 SNMP v3 Security

SNMP v3 specifications enable any two devices to communicate in a completely secure fashion using message authentication to validate users and encryption to ensure the secrecy of the communication. SNMP v3 provides a User-based Security Model (USM) to establish authentication and secrecy.

NerveCenter supports three security levels for communicating with an SNMP v3 agent:

- NoAuth/NoPriv: Passwords for authorization and privacy are not required to communicate with the agent. NerveCenter still requires the user name and context for polling.
- Auth/NoPriv: The authorization protocol and password are required to communicate with the agent. NerveCenter requires the user name, context, and authentication password for polling.
- Auth/Priv: All security parameters are required to communicate with the agent. NerveCenter requires the user name, context, and the privacy and authentication passwords for polling.

Communication between any two SNMP v3 entities takes place on behalf of a uniquely identified user within the management domain. The security level used for this communication defines the kind of security services—message authentication and encryption—used while exchanging data. NerveCenter communicates with SNMP v3 nodes on behalf of the NerveCenter poll user in the poll context. By default, the user name for MD5 authentication is NCUser, the user name for SHA-1 authentication is NCUserSHA1 and the context is NCContext, though you can change both the user names and context in NerveCenter.

If you do not specify a security level for an SNMP v3 node, NerveCenter uses a default security level of NoAuthNoPriv, which means that message authentication and encryption services are not used for data exchange with the node. You can later change the security level in NerveCenter.

Note The NerveCenter poll user, context, authentication password, and privacy password can be changed in NerveCenter Administrator. If you change the passwords, you can update this information on all nodes directly from the NerveCenter Administrator.

The security level used by NerveCenter while polling SNMP v3 nodes is configured for each node in NerveCenter Client. Information specific to nodes, such as version, security level, and authentication protocol, are entered in NerveCenter Client for the node.

NerveCenter Support for SNMP v3 Digest Keys and Passwords

SNMPv3 protocols allow any two devices to communicate in a completely secure fashion using message authentication and message encryption to ensure the secrecy of the communication. In any SNMP v3 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, called a 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, for the user.

SNMP v3 specifications have defined a localized key-generation scheme. For every user, the authentication key at every SNMP v3 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 SNMP v3 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 SNMP v3 nodes on behalf of the NerveCenter poll user (by default, NCUser for MD5 authentication and NCUserSHA1 for SHA-1 authentication) in the poll context (NCContext by default). NerveCenter needs to know the authentication and privacy passwords for this user in order to generate the keys required for secure communication. Whenever NerveCenter learns the snmpEngineID of a newly discovered SNMP v3 agent with a security level other than NoAuthNoPriv, NerveCenter generates these keys for the NerveCenter poll user on that agent. By default, the passwords are NCUserAuthPwd (authentication) and NCUserPrivPwd (privacy), though you can change both in NerveCenter Administrator. These passwords are used for all nodes that NerveCenter manages.

When the message is sent, 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. This digest 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 available in its local store to generate its local copy of the digest from the message.

- Compares the two digests (i.e. one received in the message and one generated locally). If both digests are the same, the recipient authenticates or decrypts the message using the corresponding local key. If the digests are not the same (indicating a lack of authentication), the recipient discards the message.
- The recipient reads and processes the message.

Discovery and Initialization of SNMP v3 Agents

NerveCenter can discover and monitor nodes with SNMP v1 and v2 agents without knowing anything in particular about the agent beforehand. For nodes with SNMP v3 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, a node must be initialized for NerveCenter to obtain engine information required for polling.
- When a node's version is changed to SNMP v3 from some other version, the node must be initialized.

Before NerveCenter can initialize a node, the SNMP v3 agent must be configured on the node with the poll user, poll context and security level.

- See SNMP v3 security for information about the poll user, poll context, and security level.
- See *Configuring an SNMP v3 Agent for NerveCenter* on page 108 for configuration details.

Initialization consists of the following:

- Obtaining the snmpEngineID value for the agent. To obtain this information, the agent must be configured with an "initial" user that has a security level of NoAuthNoPriv. See *Configuring an Initial User for Discovering an SNMP v3 Agent* on page 110 for details.

  The engine ID is all that's required for discovering an SNMP v3 node.

- 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 both the snmpEngineBoots and snmpEngineTime values of the SNMP v3 agent. NerveCenter must know the engine ID before it can request the boots and time information from the agent.

Note SNMPv3 specifications makes a provision for allowing completely secure initial configuration of agents wherein an agent need not disclose its snmpEngineID. Network management applications must obtain on their own the snmpEngineID of such agents. NerveCenter cannot initialize or poll such agents.

Configuring an SNMP v3 Agent for NerveCenter

Before NerveCenter can discover or poll an SNMP v3 agent on a node, the following information must be configured on the SNMP v3 agent:

- NerveCenter user (also referred to as poll user)
- NerveCenter context (also referred to as poll context)

NerveCenter then polls the SNMP v3 agent on behalf of the poll user in this poll context.

In addition to the poll user and context, the following parameters must be configured on the SNMP v3 agent.

Note The following descriptions serve only as guidelines for configuring SNMP v3 agents for NerveCenter. For exact procedural details about configuring SNMP v3 agents, consult the user documentation supplied by your SNMP v3 agent provider.

- **User Name (MD5)**: (usmUserName) NerveCenter communicates with the SNMP v3 agents using MD5 authentication on behalf of the user name you provide. By default the MD5 authentication user name is NCUser. The user name must be configured on all SNMP v3 agents using MD5 authentication with whom NerveCenter communicates. If you specify a different name for this user on the agent, then you must configure this new user name on all the SNMP v3 agents using MD5 authentication managed by NerveCenter. Moreover you must change the name accordingly in NerveCenter Administrator. The MD5 authentication user name and the SHA-1 authentication user name must be different.
- **User Name (SHA-1)**: (usmUserName) NerveCenter communicates with the SNMP v3 agents using SHA-1 authentication on behalf of the user name you provide. By default the SHA-1 authentication user name is NCUserSHA1. The user name must be configured on all SNMP v3 agents using SHA-1 authentication with whom NerveCenter communicates. If you specify a different name for this user on the agent, then you must configure this new user name on all the SNMP v3 agents using SHA-1 authentication managed by NerveCenter. Moreover you must change the name accordingly in NerveCenter Administrator. The MD5 authentication user name and the SHA-1 authentication user name must be different.
- **User Context**: (vacmContextName) User context in which NerveCenter communicates with the SNMP v3 agent. NerveCenter uses only one user context for communicating with all SNMP v3 agents, which by default is NCContext. NCContext must be configured on all SNMP v3 agents with whom NerveCenter communicates. If you specify a different context name on the agent, then you must configure this new user context on all the SNMP v3 agents managed by NerveCenter. Moreover you must change the name accordingly in NerveCenter Administrator.

  When determining the context permissions for the NerveCenter user, make sure the user can access all the MIBs that you want NerveCenter to manage. That means the NerveCenter user/context should have access to all the MIB objects that you want NerveCenter to manage, given the security level you choose for the agent.

- **Security level**: (SecurityLevel) Security level NerveCenter uses to communicate with the SNMP v3 agent. NerveCenter supports all three security levels specified by SNMP v3: authPriv, noAuthNoPriv, and authNoPriv. If you later change the security level for the agent on the node, you must also change this information for the node in NerveCenter Client.

  Make sure to specify the appropriate level of security for the information NerveCenter manages. If you specify noAuthNoPriv, you will not have authentication or encryption for your SNMP v3 messages.

- **Authentication Protocol**: (usmUserAuthProtocol) At present SNMP v3 specifications define two message authentication schemes: HMAC-MD5-96 (MD5) and HMAC-SHA-96 (SHA). You can choose either of these two protocols for the agent, but you must specify the protocol you choose in NerveCenter Client for the respective node. By default, NerveCenter specifies MD5 for all nodes. If you choose to configure the agent with SHA instead, then you must change this information for the node in NerveCenter Client. The key localization scheme defined in SNMP v3 specifications has a link with the authentication protocol. Therefore, whenever you change the authentication protocol, NerveCenter must recompute the keys for the agent concerned. If you are using noAuthNoPriv security level, then configure the SNMP v3 agent with usmNoAuthProtocol to be used for communication. Of course, then you will not have either authentication or encryption for your SNMP v3 messages.

- **Privacy Protocol**: (usmUserPrivProtocol) SNMP v3 standard specifies that CBC-DES can be used for encryption services, and that is the only protocol NerveCenter supports. If you want encryption services, then configure the SNMP v3 agent to use CBC-DES as its privacy protocol to communicate on behalf of NerveCenter user in context with the AuthPriv security level. If you are using noAuthNoPriv or AuthNoPriv security levels, then you can configure the SNMP v3 agent with usmNoPrivProtocol. Of course, then you will not have encryption for your SNMP v3 messages. If you use some protocol other than CBC-DES, NerveCenter can not use encryption services to manage the agent.

- **Passwords**: (AuthKey, PrivKey) Configuring the NerveCenter user also involves configuring the authentication and privacy passwords. The passwords you provide are converted to authentication keys at runtime.

  NerveCenter uses the following as default passwords for the NerveCenter user:

  - NCUserAuthPwd (for the authentication password)
  - NCUserPrivPwd (for the privacy password)

  If you choose a security level of authPriv for a node, both passwords are used. For a security level of authNoPriv, only the authentication password is used. For a security level of noAuthNoPriv, no passwords are used. Depending on the security level you set up for the agent, you must configure the NerveCenter user with these same passwords:

  These passwords, along with snmpEngineID and the authentication protocol (MD5 or SHA) are used to generate keys (AuthKey and PrivKey respectively). Refer to the SNMP v3 specifications for more information. If you choose different authentication and privacy passwords in NerveCenter, then configure all SNMP v3 agents managed by NerveCenter to

use the authentication and privacy passwords designated for the NerveCenter user. If you change the passwords on one agent, you must make the change on all SNMP v3 agents managed by NerveCenter.

Initially you need to configure these passwords directly on all SNMP v3 agents that you want NerveCenter to manage. If you later change the passwords, you can do so from the NerveCenter Administrator.

Configuring an Initial User for Discovering an SNMP v3 Agent

To discover an SNMP v3 node, NerveCenter sends a GetRequest message to that node with the following parameters:

- SecurityLevel: set to (noAuthNoPriv)
- MsgUserName: set to (initial)
- MsgAuthoritativeEngineID: set to zerolength (‘ ‘)
- Empty variable bindings (variable-bindings=‘ ‘)

Note For exact procedural details about configuring SNMP v3 agents, consult the user documentation supplied by your SNMP v3 agent provider.

A properly configured SNMP v3 agent responds to the GetRequest with a Report PDU, which includes its local snmpEngineID (supplied within the msgAuthoritativeEngineID field). After NerveCenter obtains the snmpEngineID, NerveCenter can discovers the node as an SNMP v3 agent.

This process of discovering SNMP v3 agents is recommended in SNMP v3 specifications. Most of the SNMP v3 agents are preconfigured with an “initial” user and respond in the manner previously described to the GetRequest, so you generally won’t have to perform any special type of configuration on your agent. Refer to the user documentation supplied by your SNMP v3 agent provider for detailed information.

SNMP Auto and Manual Classification Settings

Using manual or auto-classification, NerveCenter can send requests to determine the highest SNMP version 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, either supplied manually by the user or obtained via classification, before NerveCenter can poll the node or process a trap from the node.

More information is contained in the following sections:

- *How NerveCenter Classifies a Node’s SNMP Version* on page 111

How NerveCenter Classifies a Node's SNMP Version

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 *Designing and Managing Behavior Models*.
- 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).

For a detailed study of classification, refer to the white paper "Open NerveCenter: Node Classification," which ships with the NerveCenter online guides. Following is a summary of classification.

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

While classifying a node, NerveCenter attempts to detect the maximum supported version on the agent up to a maximum specified version, which you can configure in NerveCenter Administrator. So, 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, any existing version information for the node is lost during classification. For example, if the node was previously identified as SNMP v3 and is now changed (to v1, v2c, or Unknown), then the v3 related security information for the node is lost.
- If NerveCenter fails to classify the node, then the version of the node is set to "Unknown." NerveCenter cannot poll a node with an unknown version.
- A node must have correct version information, either supplied manually by the user or obtained via classification, before NerveCenter can poll the node or process a trap from the node.

When NerveCenter Classifies a Node's SNMP Version

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 *Designing and Managing Behavior Models*.
- Automatically—You can set up auto-classification in NerveCenter Administrator. Then, 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 auto-classification in the following instances:

- A node is added through a node file either 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 another NerveCenter Server, and the node does not have a version or has the version "Unknown."
- A node is added from a trap, and the node's version is not v3. NerveCenter needs to verify whether these nodes are v1 or v2. If the trap indicates v3, NerveCenter does not need any further verification.
- NerveCenter is co-resident with network management platform and the platform sends nodes to NerveCenter. All nodes added from OpenView Network Node Manager are v1 by default.

Note NerveCenter does not attempt auto-classification for nodes that you add manually in Client.

Disabling auto-classification in Administrator prevents auto-classification for all these cases. If you choose to disable auto-classification, bear in mind that NerveCenter does not poll nodes whose SNMP version is unknown. (You can still classify nodes manually in NerveCenter Client using the available commands.)

Enabling Auto-classification

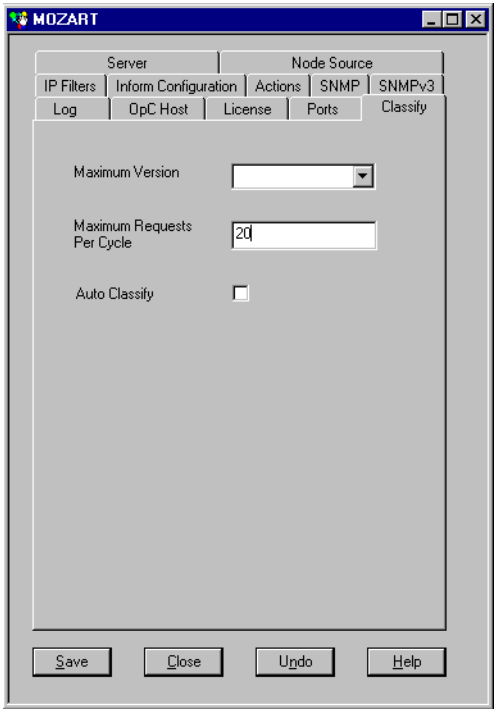
NerveCenter can be configured to send a request that determines the highest SNMP version on each node automatically when the node is added to the NerveCenter database. You specify separately the highest version you want NerveCenter to detect. See *Setting a Maximum Classify Value* on page 114.

❖ To enable auto-classification and 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 38.

2. Select the Classify tab.

   The Classify tab is displayed.



3. Select the AutoClassify checkbox.

   NerveCenter auto-classification is enabled up to the version you select.

Note To disable auto-classification, deselect the AutoClassify checkbox. If you disable auto-classification, bear in mind that NerveCenter does not poll nodes whose SNMP version is unknown.

4. Select Save.

Setting a Maximum Classify Value

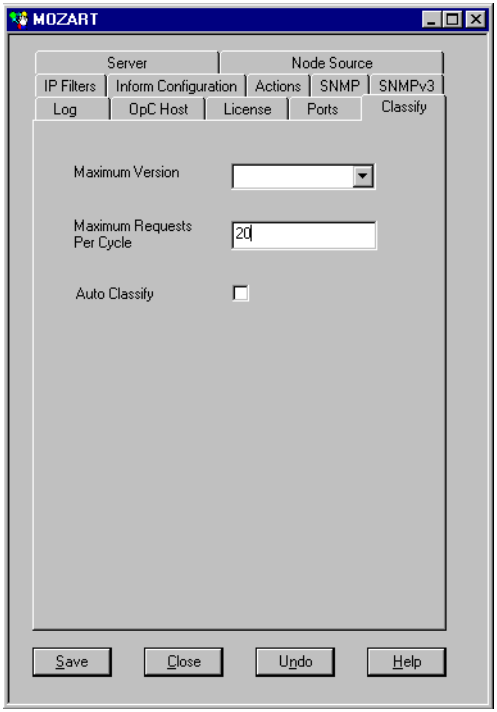
Whether using auto-classification or manual node classification (in Client), you specify the highest version you want NerveCenter to detect. When performing classification, NerveCenter will never attempt to classify a node for any version above the version you specify. For example, if you select v2c, NerveCenter can send classification requests only for SNMP v1 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 38.

2. Select the Classify tab.

   The Classify tab is displayed.



3. Select the highest level you want to classify from the Maximum Version drop-down listbox.

4. Select Save.

Setting the Maximum SNMP v3 Requests per Cycle

NerveCenter allows you some control over SNMP v3 overall performance by setting the maximum number of SNMP v3 requests per processing cycle. An SNMP v3 processing cycle occurs approximately every second.

The value you enter determines the number of requests NerveCenter processes simultaneously for all v3 operations including classification, SNMP Test Version poll, version change to or from v3, authentication protocol change, security level change, initialization requests, and others.

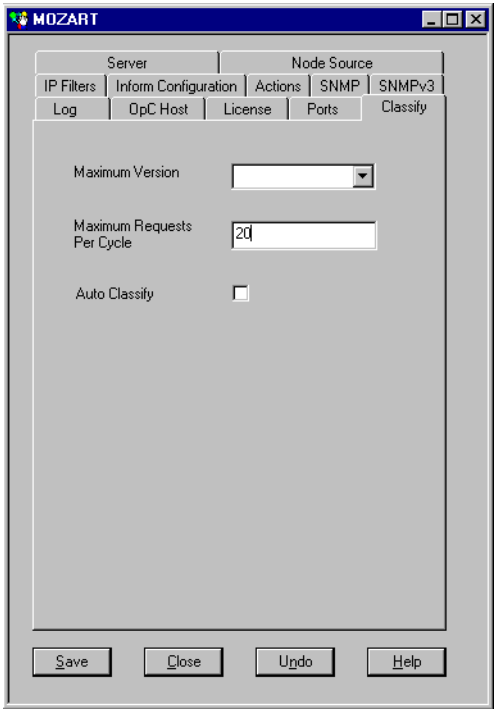
When deciding the rate of v3 requests, you should consider how many SNMP messages per second your network devices can and should handle. Also consider whether the messaging uses authentication and privacy keys, which can slow down performance even more.

❖ To set the maximum number of SNMP v3 requests:

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server. See *Connecting Administrator to a NerveCenter Server* on page 38.

2. Select the Classify tab.

The Classify tab is displayed.



3. In the Maximum Requests Per Cycle field, enter the maximum number of SNMP v3 requests you want NerveCenter to process. The default value is 20.

4. Select Save.

NerveCenter will handle SNMP data according to these settings.

SNMP v3 Security Settings

Before NerveCenter can poll SNMP v3 agents, the agents must be configured to support a NerveCenter user. By default the user name is NCUser, though you can change the name in NerveCenter Administrator.

The NerveCenter user (NCUser) shall have a context (NCContext by default) and two passwords (authentication and privacy). You must supply the authentication and privacy passwords for the NerveCenter user configured on your devices. The security level and authentication protocol are set at the node level in NerveCenter Client.

More information is contained in the following sections:

- *Changing the SNMP v3 Key Passwords* on page 117
- *Changing the NerveCenter SNMP v3 User Name and Context* on page 119

Changing the SNMP v3 Key Passwords

The authentication and privacy protocols require specialized keys, called authentication and privacy keys. These keys are generated from corresponding passwords. Before NerveCenter can poll SNMP v3 nodes, NerveCenter requires passwords for the NerveCenter user (NCUser) configured on your v3 agents.

By default, these passwords are as follows when you install NerveCenter.

- The default authentication password is NCUserAuthPwd. This password is required for AuthNoPriv and AuthPriv security.
- The default privacy password is NCUserPrivPwd. This password is required for AuthPriv security.

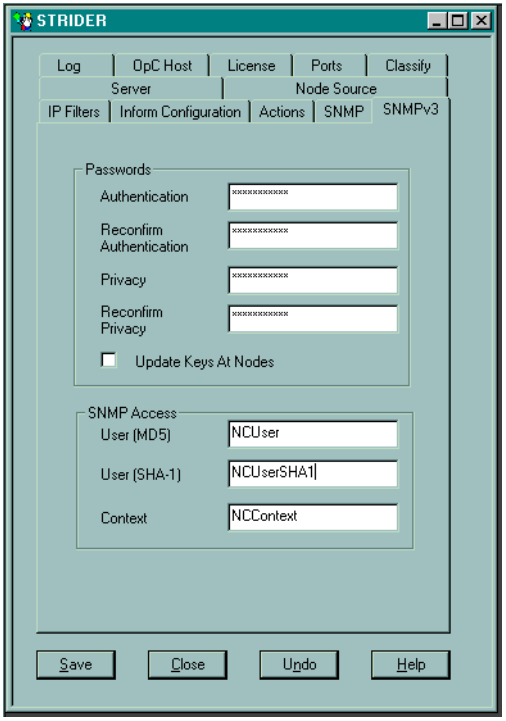
After installation, you can change the passwords as appropriate for your network management strategy. Optionally, you can command NerveCenter to update the new passwords on all managed nodes.

Note Each password must have at least eight characters, with a maximum length of 32 characters. The passwords are case sensitive.

❖ To provide or change SNMP v3 passwords:

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server. See *Connecting Administrator to a NerveCenter Server* on page 38.
2. Select the SNMPv3 tab.

The SNMPv3 tab is displayed.



3. Enter your authentication password in the Authentication field.
4. Retype this password in the Reconfirm Authentication field.
5. Enter your privacy password in the Privacy field.
6. Retype this password in the Reconfirm Privacy field.
7. To update the passwords on all managed agents, select Update Keys At Nodes.

   You would normally want NerveCenter to update the passwords on your agents. If you prefer to do this manually or if you have already set the passwords on each agent, then do not select the checkbox.
8. Select Save.

   A message box displays a warning that NerveCenter must stop polling SNMP v3 nodes while performing this operation.
9. Select Yes to save the new passwords or No to cancel the change.

When NerveCenter updates passwords on managed agents, all polling of SNMP v3 nodes is paused until the operation is complete. In addition, various v3 operations (e.g., change of authentication protocol or security level) are not available in NerveCenter Client during this operation.

After the key change, NerveCenter is configured with the new passwords even if the key change operation fails on an agent. Key change failure is indicated in the node's properties in NerveCenter Client.

If NerveCenter Server is stopped during a remote key change operation, the operation resumes automatically when the Server is restarted.

Changing the NerveCenter SNMP v3 User Name and Context

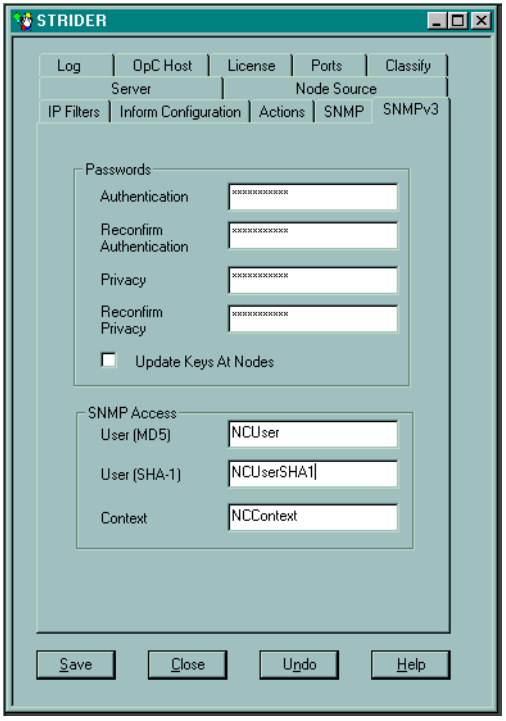
Before NerveCenter can poll SNMP v3 agents, the agents must be configured to support a NerveCenter user and context. By default the MD5 user name is NCUser, the SHA-1 user name is NCUserSHA1, and the context is NCContext. If you change these values on your agents, you must make the same change in NerveCenter Administrator. You may want to change the user names and context on a regular basis to ensure privacy, or if you have multiple NerveCenter managing different categories of devices, you may want different user names and context associated with each category.

❖ To change the user name or context:

1. Open NerveCenter Administrator and connect to the appropriate NerveCenter Server. See *Connecting Administrator to a NerveCenter Server* on page 38.

2. Select the SNMPv3 tab.

The SNMPv3 tab is displayed.



3. Enter enter the new MD5 user name in the User (MD5) field. Select the existing text and type the name you want.

4. Enter enter the new SHA-1 user name in the User (SHA-1) field. Select the existing text and type the name you want.

Caution The MD5 authentication user name and the SHA-1 authentication user name must be different.

5. Enter the new context in the Context field.

6. Select Save.

NerveCenter is configured with the new user name and context and will use these names when communicating with managed nodes.

SNMP v3 Operations Log

Whenever a NerveCenter Server receives a request for an SNMP v3 operation (e.g. authorization or privacy key change request) or an error occurs while attempting to perform an SNMP v3 operation (e.g. v3 initialization fails), the NerveCenter Server logs a message to a file. This log file, named `v3messages.log`, resides in the NerveCenter installation log directory on the NerveCenter Server host machine. The file contains messages about SNMP v3 operations and errors resulting from requests that originate with any connected NerveCenter Clients, Administrators, and Command Line interfaces.

When an error occurs after attempting to perform an SNMP v3 operation, aside from logging the error in the log file, 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 a dialog box with the error that is logged.
- If you are logged on to some other NerveCenter Client or Administrator (one that did not initiate the error condition), you see a red icon in the status bar. When you double-click the icon, a dialog box displays the NerveCenter Server with the SNMP v3 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.

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 this fact in the log file and changes the red icon to green for all connected Clients and Administrators.
- Dismiss the dialog box without acknowledging the error condition. If you merely dismiss the dialog box, only the icon in your Client or Administrator 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, or unsigned. Moreover, the Server does not indicate acknowledgment in the log file.

If the SNMP v3 operation affects a group of nodes (e.g., version change or classification failure), you will see only one instance for the group displayed in the error message dialog box. To see details for each node, you can look in the log file.

Whether you acknowledge or dismiss the error, all messages remain in the `v3messages.log` for you to read.

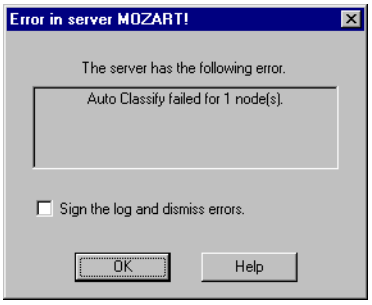
For more information, refer to the following topics:

- *Signing a Log for SNMP v3 Errors Associated with Your Administrator* on page 122
- *Signing a Log for SNMP v3 Errors Associated with a Remote Client or Administrator* on page 123
- *Viewing the SNMP v3 Operations Log* on page 124

Signing a Log for SNMP v3 Errors Associated with Your Administrator

Whenever an SNMP v3 operation is requested or an error occurs while attempting an SNMP v3 operation, the NerveCenter Server logs a message to a file. If you are logged in to the NerveCenter Administrator that initiated the request causing a logged condition, NerveCenter displays a dialog box with the error that is logged.

Figure 8-1. Operations Log Error in Server Dialog Box for Your Administrator



You can acknowledge a logged condition from NerveCenter Administrator 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.

❖ 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. Select 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 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:

- Select OK without checking the checkbox.

  In this case, only the icon in your Administrator 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 SNMP v3 Errors Associated with a Remote Client or Administrator

Whenever an error occurs while attempting an SNMP v3 operation, the NerveCenter Server logs a message to a file. If you are logged on to some remote NerveCenter Administrator (one that did not initiate the error condition), you see a red icon in the status bar.

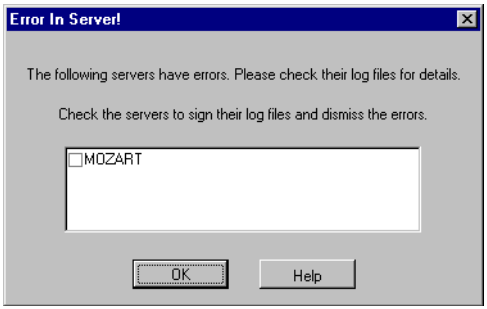
You can acknowledge a logged condition from NerveCenter Administrator by signing the Operations log. Signing the log causes the icon to turn green in all connected Clients/Administrators.

You can also dismiss the dialog box without acknowledging the error condition.

❖ To sign the Operations log:

1. Double-click the red icon in the Status Bar.

   The Error In Server dialog box is displayed.



2. Check the NerveCenter Server or Servers for which you want to sign the log.

3. Select 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. Select OK without checking any of the checkboxes.

In this case, only the icon in your Administrator 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 SNMP v3 Operations Log

Whenever an SNMP v3 operation is requested or an error occurs while attempting the operation, the NerveCenter Server logs a message to a file. This log file, named `v3messages.log`, 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 8-1. Fields in the Operations Log

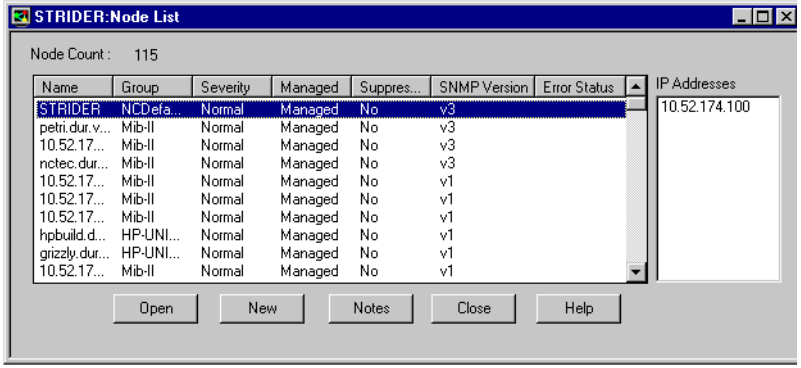
Field	Description
Date/Time	Date and time the record was logged. The format is month/day/year, hour/minute/second, and day (for example, 12/16/2000 11:32:29 Sat).
EventID	This always NC_SERVER.
CategoryID	Name of the thread where the event occurred.
Error Status	One of several error status strings See *SNMP Error Status* on page 125 for a description of SNMP v3 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).

The following illustration shows the Node List window in the Client.

Figure 8-2. Node List Window



Though most of the error strings correspond to SNMP v3 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.

- **AuthKeyFail** – The change for the authentication key failed. Polling will not happen for nodes with this error. You must rectify the problem manually on the agent and use the Test Version poll to verify NerveCenter communication with this node.
- **PrivKeyFail** – The change for the privacy key failed. Polling will not happen for nodes with this error. You must rectify the problem manually on the agent and use the Test Version poll to verify NerveCenter communication with this node.
- **AuthPrivKeyFail** – Change for both the authentication and privacy keys failed. Polling will not happen for nodes with this error. You must rectify the problem manually on the agent and use the Test Version poll to verify NerveCenter communication with this node.
- **V3InitFail** – An attempt to get the engine ID failed and NerveCenter could not initialize the node. Polling will not happen for this node. You can try running the Test Version poll, which attempts to get the engine ID for this node again. Alternatively, if the node sends a trap that NerveCenter can decode, NerveCenter will then get the engine ID from that trap.

- **ClassifyFail** – At attempt to obtain the node's version failed during a classification attempt. The version will be "Unknown" for this node and polling will not happen. You can manually change the version or try to classify the node again.
- **AutoClassifyFail** – At attempt to obtain the node's version failed during a classification attempt while NerveCenter was using auto-classification. The version will be "Unknown" for this node and polling will not happen. You can manually change the version or try to classify the node again.

Note ClassifyFail and AutoClassifyFail status values are not limited to SNMP v3 agents. If NerveCenter attempts classification of an agent and the classification attempt 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 SNMP v3 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 NerveCenter is 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 the decoding of traps always results in a Configuration Mismatch error message.

- **TimeSyncFail** – An attempt to get the engine boots/timeticks failed for the node. 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. Moreover, it is easy to recover from this error status. Right-click the node in the Node List and select v3TestPoll. If the agent corresponding to the node is up, the test poll should be successful and NerveCenter will clear the error message.

NerveCenter will not poll any nodes whose error status is one of the following:

- AuthKeyFail
- PrivKeyFail
- AuthPrivKeyFail
- TestVersionFail
- V3InitFail
- ClassifyFail

Managing NerveCenter Security 9

NerveCenter security allows access for two different groups. A NerveCenter administrator must add users to an appropriate user group before they can use NerveCenter.

Note On Windows, NerveCenter can be installed without security. If NerveCenter is installed without security, 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 local machine or an NIS database. Membership in the appropriate groups determines what privileges users have within NerveCenter.

Note The user ID under which the NerveCenter Server runs on Windows must be a member of the NerveCenter Admins group. On UNIX, root must run the server; however, after the server initializes, control is given to the nervectr user, which is a member of ncadmins.

Members of the ncadmins group on UNIX or the NerveCenter Admins group on Windows can configure NerveCenter and create and modify behavior models. Members of the ncusers group on UNIX or the NerveCenter Users group on Windows can monitor network status and reset alarms. For more details about privileges, see the section *NerveCenter Login Rights* on page 28.

This chapter includes the following sections:

Section	Description
Managing Security on UNIX on page 130	Explains how to add users to user groups on UNIX.
Managing Security on Windows on page 130	Explains how to add users to user groups on Windows.

Managing Security on UNIX

During a NerveCenter installation, the ncadmins and ncusers groups and the nervectr user, which is a member of ncadmins, are created. After installation, a NerveCenter administrator must add users to the appropriate groups.

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 30.
- Add NerveCenter users to the ncusers group. NerveCenter users will only use the NerveCenter Client to monitor networks.

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 two scenarios for adding NerveCenter users:

- *Adding Users to a NerveCenter Installed on a Windows Workstation or Server* on page 131
- *Adding Users to a NerveCenter Installed on a Windows Domain Controller* on page 132

Adding Users to a NerveCenter Installed 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. If NerveCenter is installed instead on a Windows domain controller, see *Adding Users to a NerveCenter Installed on a Windows Domain Controller* on page 132.

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 30.
- Add NerveCenter users to the ncusers group. NerveCenter users will only use the NerveCenter Client to monitor networks. See *NerveCenter Login Rights* on page 28 for a more detailed discussion of permissions.

Remember, once you add users to a group, they must log out and log 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 Administrator 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 Administrator 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 your 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 Installed 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. If the NerveCenter Server is on a Windows workstation or server instead, see *Adding Users to a NerveCenter Installed on a Windows Workstation or Server* on page 131.

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 30.
- Add NerveCenter users to the ncusers group. NerveCenter users will only use the NerveCenter Client to monitor networks. See *NerveCenter Login Rights* on page 28 for a more detailed discussion of permissions.

Remember, once you add users to a group, they must log out and log 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 Administrator 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.

Troubleshooting: Managing NerveCenter Security

The following list contains some common security problems NerveCenter users face.

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 130 or *Managing Security on Windows* on page 130.

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 3, *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 130 or *Managing Security on Windows* on page 130.

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.

Managing NerveCenter Alarm Actions 10

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 138	Explains how to configure NerveCenter to send e-mail notifications via an SMTP server.
Configuring NerveCenter to Send Microsoft Mail on page 139	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 145	Explains how to configure NerveCenter on Windows to send pages.
Setting Dialing Properties for Paging Notification on UNIX on page 146	Explains how to configure NerveCenter on UNIX to send pages.
Specifying Settings for Log Management on page 148	Explains how to configure NerveCenter to manage logs.
Troubleshooting: Managing Alarm Actions on page 151	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 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.
- With a Microsoft Exchange server. See *Configuring NerveCenter to Send Microsoft Mail* on page 139.

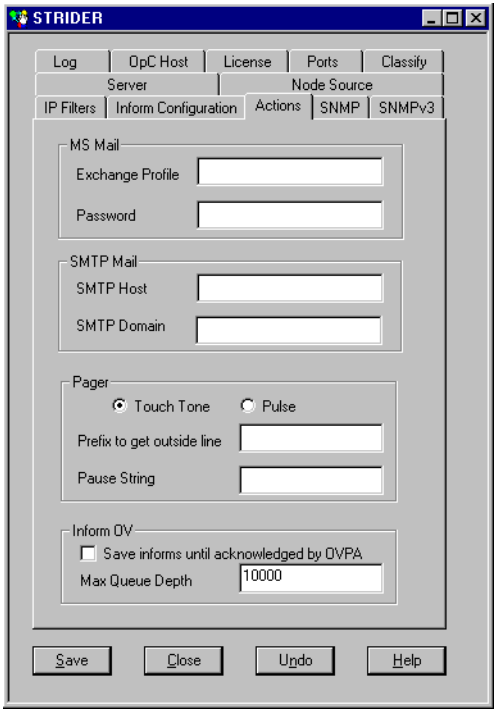
The recipient or recipients of the SMTP mail message is specified when the behavior model is created. However, before this message can be sent, the administrator must specify which SMTP server NerveCenter uses to send the message.

❖ 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 38.

2. Select the Actions tab.

   The Actions tab is displayed.



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 138.)
- 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 NerveCenter Server as a Service* on page 140
- *Configuring NerveCenter to Send Microsoft Mail while Running NerveCenter Server as an Process* on page 143

Caution There are other scenarios in which the Microsoft Mail action can be used without creating a global NerveCenter group or account. However, other scenarios involve switching Exchange profiles every time a new user logs on to the machine running NerveCenter.

Configuring NerveCenter to Send Microsoft Mail while Running NerveCenter Server 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. To configure NerveCenter to send Microsoft mail properly, you must have a network administrator create a separate NerveCenter account and Exchange mailbox.

❖ To configure NerveCenter to send Microsoft mail while running NerveCenter Server as a service:

1. 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.

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.

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.

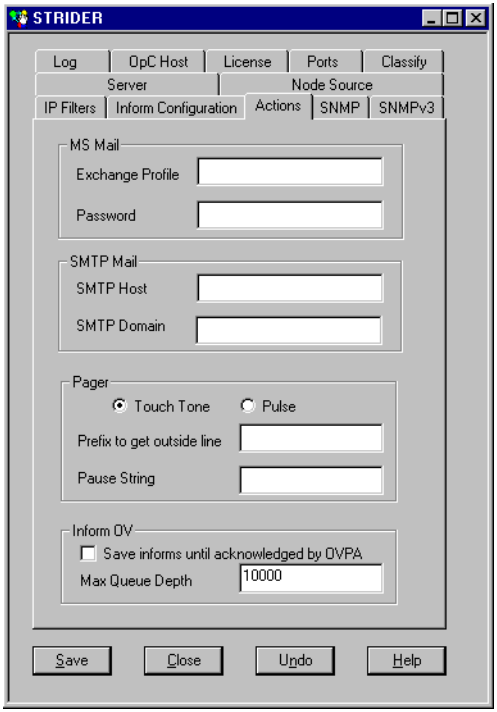
4. On the machine that will run NerveCenter Server, create a NerveCenter-designated Exchange profile to point to the machine hosting the Microsoft Exchange Server and the NerveCenter-designated mailbox created in step 1.

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 38.

6. Select the Actions tab.

The Actions tab appears.



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 NerveCenter Server as an 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 NerveCenter Server as a Service* on page 140.

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.

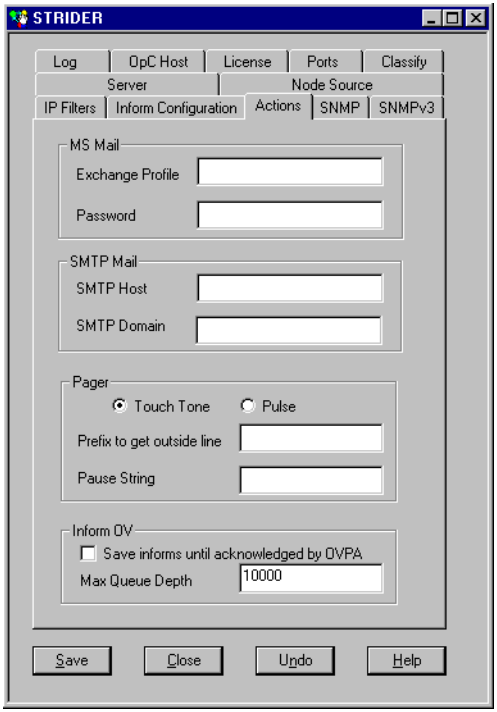
3. On each machine that will run NerveCenter Server, create a NerveCenter-designated Exchange profile to point to the machine hosting the Microsoft Exchange Server and the NerveCenter-designated mailbox created in step 1.

   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 38.

5. Select the Actions tab.

The Actions tab appears.



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 call the necessary person's pager.

Data about the recipient's pager is specified when the behavior model is created. However, before the page can be sent, an administrator must configure basic settings concerning the server's dialing properties.

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 146.

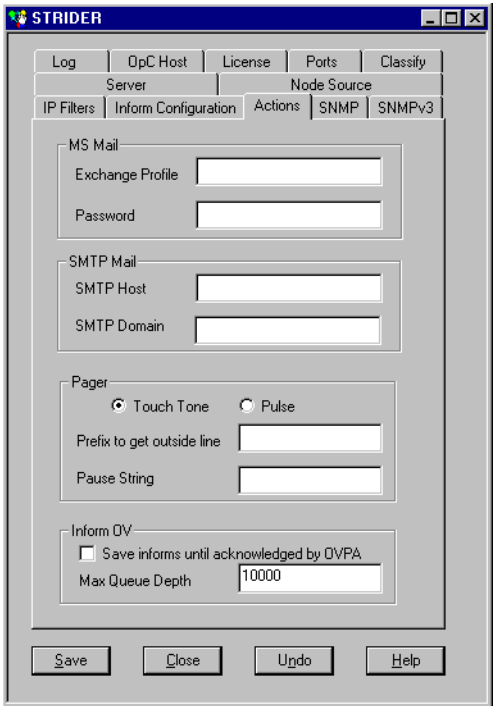
Note A paging NerveCenter Server must have access to a TAPI-compliant modem and access to an outside line.

❖ 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 38.

2. Select the Actions tab.

   The Actions tab is displayed.



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 call the necessary person's pager.

Data about the recipient's pager is specified when the behavior model is created. However, before the page can be sent, an administrator must configure basic settings concerning the server's dialing properties.

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 145.

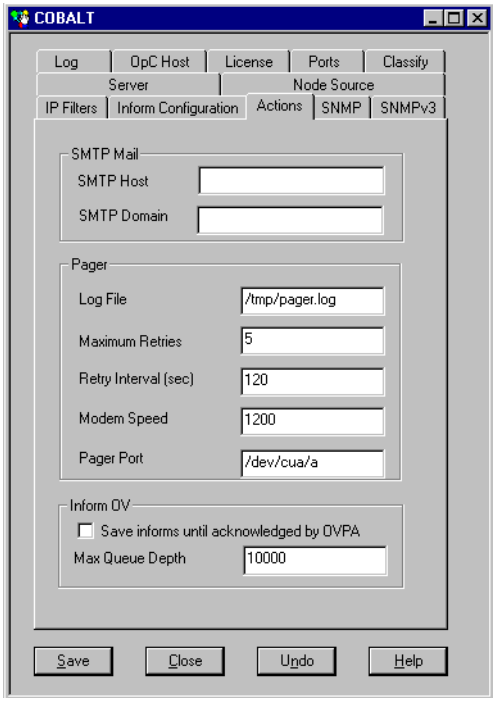
Note A paging NerveCenter Server must be on a machine with a modem and access to a outside line.

❖ 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 38.

2. Select the Actions tab.

The Actions tab is displayed.



3. In the Log File field, type the full path and name of the pager log file.

   The Pager action logs messages about success or failure to this file. Each log entry contains the following:

   - a date and time stamp
   - the process ID (PID) of the Pager process
   - the log message itself

   If, for some reason, 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 you want NerveCenter to wait between dialing.

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=`

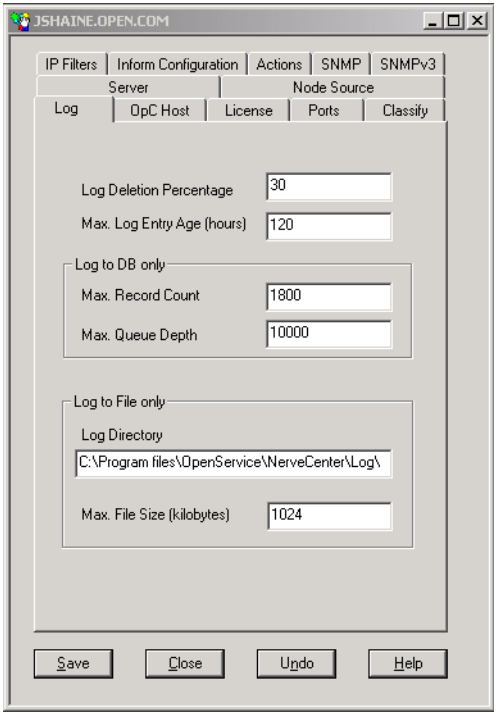
  nonverbose: `11/11/2002 15:14:43 Mon;4875;Normal;Mib-II;MyComputer;AllTraps_LogToFile;Logging;allTraps; Logging;321;4;1.3.6.1.41.78;0;;`

❖ 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 38.

2. Select the Log tab.

   NerveCenter displays the Log tab.



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 138.

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 139.

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 145 or *Setting Dialing Properties for Paging Notification on UNIX* on page 146.

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 148.

Managing NerveCenter Web Integration 11

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 154	Explains how to start and stop the NerveCenter Web Collector to allow for web integration.

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 154.
- *Starting and Stopping the NerveCenter Web Collector in Windows* on page 154.

Starting and Stopping the NerveCenter Web Collector in UNIX

During a typical installation, the NerveCenter Web Collector is installed as a daemon. Therefore, whenever you boot the host machine, the NerveCenter Web Collector will automatically start.

However, there may be times when you want to start or stop the NerveCenter Web Collector manually.

- 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, whenever you boot the host machine, the NerveCenter Web Collector will automatically start.

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.

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 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 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\CGI.

   c. In the Alias field, type **`NerveCenterCGI`**.

   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 have 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 NerveCenterCGI.

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 156.

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 *NerveCenterInstallationPath*`\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 *NerveCenterInstallationPath*`\Web\CGI`.

   You may also need to create virtual directories for the subdirectories in *NerveCenterInstallationPath*`\Web` (`Conf` and `Help`)

Note If you have another web server running (Netscape FastTrack, for example), you must shut down the other web server before you can start IIS.

❖ To configure an Apache web server:

Edit the httpd.conf file as follows:

1. Edit the file to use port 80.

2. Add the following lines (for UNIX, change C:/Program Files/OpenService/ 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.

Managing the Database 12

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 If you want to install a new database, see *Installing NerveCenter* in the NerveCenter documentation. You must be a Windows administrator or UNIX root to run SerializeDB.

This chapter includes the following sections:

Section	Description
Database Formats on page 160	Describes the NerveCenter database formats on each platform.
Viewing Information about the Database on page 160	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.
Backing up the Database on page 162	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 165	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 169	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 170	Describes how to use the ncdb2html.pl script to convert the NerveCenter database to HTML.
Troubleshooting: Managing the NerveCenter Database on page 174	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 you find 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* in the NerveCenter documentation for instructions on creating a new database. Also see *Transferring Data between Databases* on page 169 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.

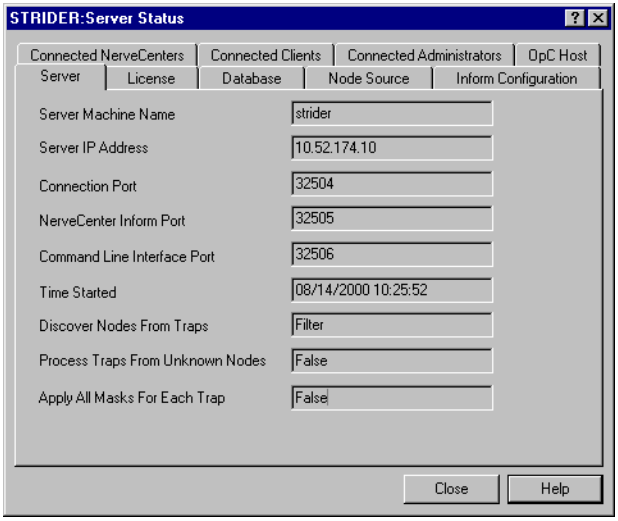
Information about the database is useful when you are troubleshooting or when you just want to verify the database information—for example, to make sure you are going to back up the contents of the correct database.

❖ 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.

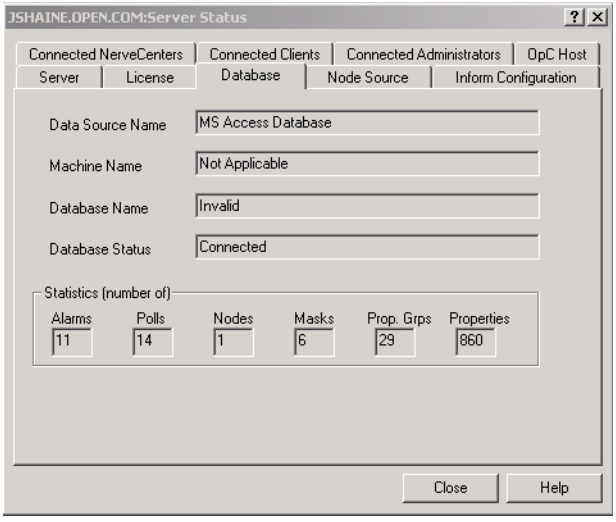
2. From the Server menu, select Server Status.

   The Server Status window is displayed.



3. Select the Database tab.

   The Database tab is displayed.



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.

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 Using the SerializeDB Application* on page 163.

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.

Backing up the Database Using the SerializeDB Application

SerializeDB 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.

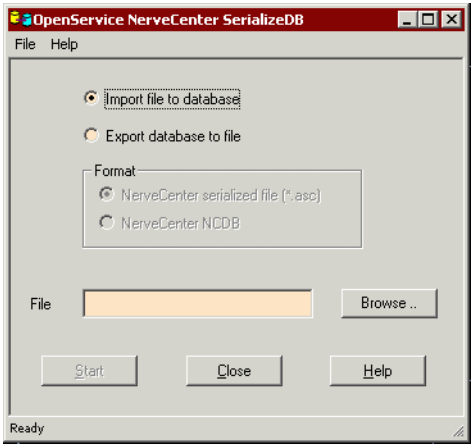
Caution Before you back up the database, you must shut down the NerveCenter Server. Not shutting down the NerveCenter Server, could result in lost data.

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 160 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 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 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)

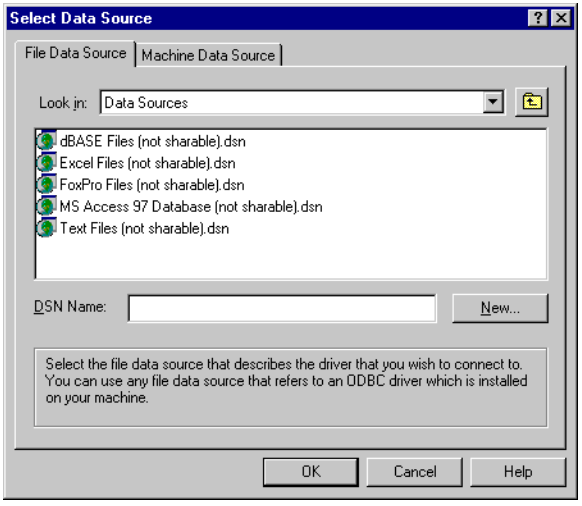
Note Unless you plan on using the resulting files directly with a UNIX server, we recommend you select the NerveCenter serialized file option. The ASC file is smaller, and because it is a single file, it is easier to manage.

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 write over existing data, select the Browse button. Select the existing file. Then select 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.



   a. Select the Machine Data Source tab.

   b. Select the data source for your NerveCenter database. Then select OK.

   See your Microsoft ODBC documentation for more information about selecting data sources.

When the data has been exported, SerializeDB displays a message letting you know that the data transfer was successful.

When running SerializeDB from the command line, you can only export to a serialized file (`.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 165
- *Restoring the Database from the Command Line* on page 167

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.

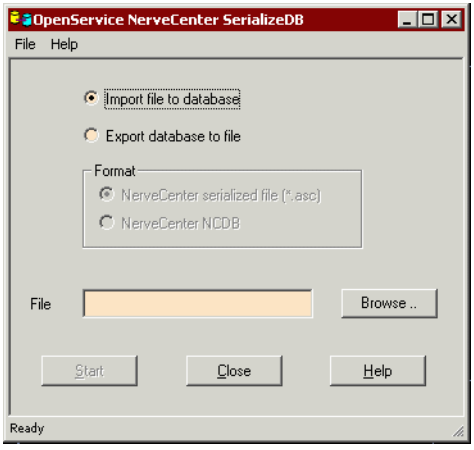
Caution Before you restore the database, you must shut down the NerveCenter Server. If you don't shut down the NerveCenter Server, you could lose data.

If you are restoring a UNIX flat file database, all of the files that make up the database must be in the db directory. See *Database Formats* on page 160 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.

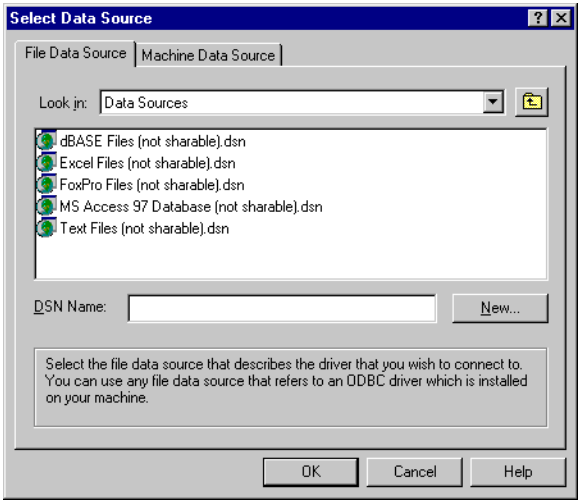
   You can also use the Browse button to select a file. In the Select File dialog box, select the file; then select Open.

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.



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 160 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|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 database or a Microsoft 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 when running in command line mode. Therefore, do not run SerializeDB remotely or on a headless server via a 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 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.

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 162)
- Import, or restore, data (*Restoring the Database* on page 165)

Caution Note to MS-Access Database Users
You must serialize your database before installing NerveCenter 3.8. You cannot use the NerveCenter 3.8 version of SerializeDB to serialize older databases.
If you did not serialize your old database, you must re-install your old version of NerveCenter and serialize your database.

To serialize a NerveCenter 3.8 database after installing NerveCenter 3.8, run SerializeDB.exe using NCAccess.mdb as the database to produce a new .asc file.

ncdb2html.pl

Included in NerveCenter 3.8 is a utility that will convert *.ncdb and *.node files to HTML, so that you can easily know what NerveCenter objects are in your database.

Figure 12-1. Sample ncdb2html.pl Output

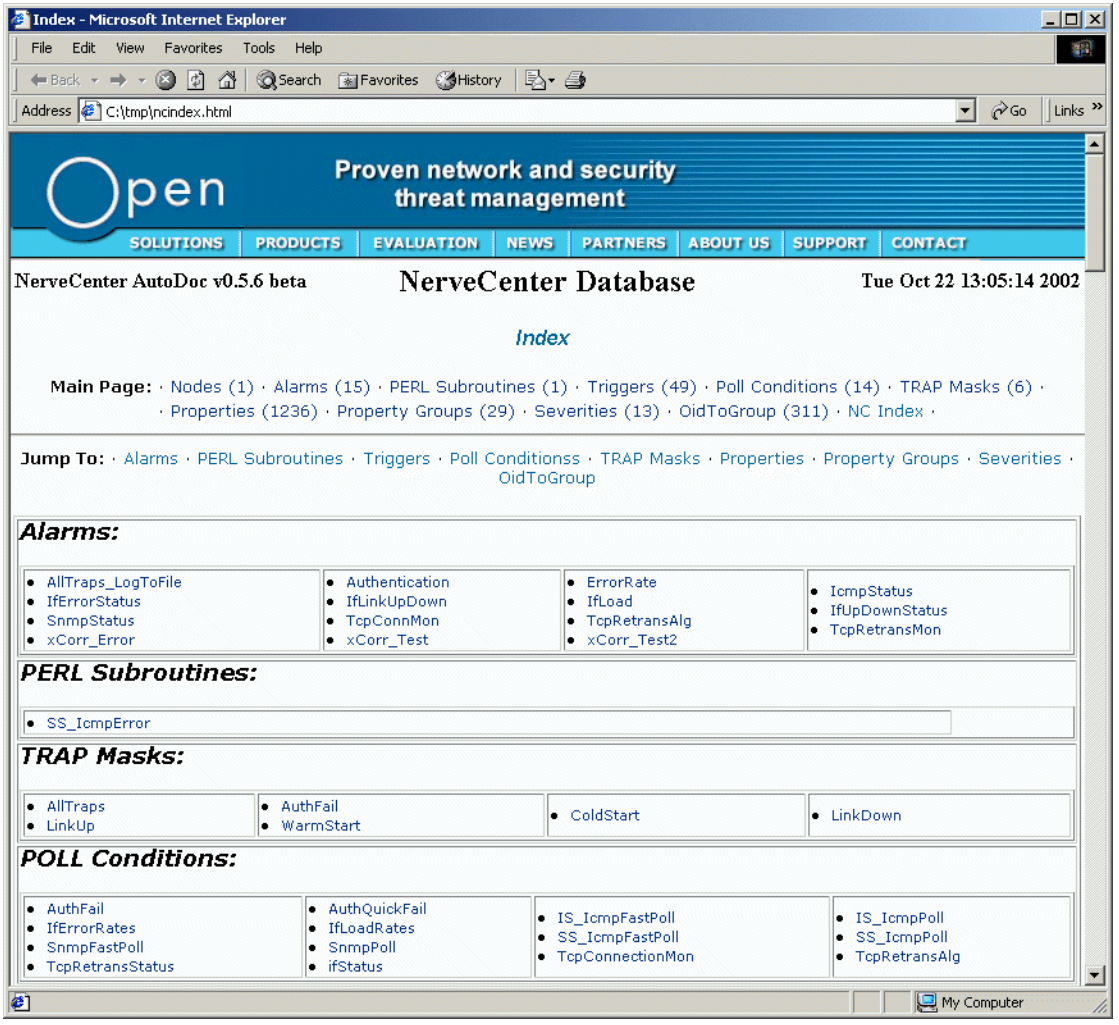


Table 12-1 details the contents of the HTML output:

Table 12-1. 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

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.

Table 12-1. Output of `ncdb2html.pl` (continued)

File	Contents
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.
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 SerialzeDB utility. For more information about SerializeDB, see *Managing NerveCenter* 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 the *installation*`/Samples/ncdb2html` directory.

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 12-2:

Table 12-2. `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 165.

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 Using the SerializeDB Application* on page 163 or *Restoring the Database from the Command Line* on page 167.

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 159.

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 162 or *Restoring the Database* on page 165.

Managing Management Information Bases (MIBs) 13

By modifying the Open NerveCenter MIB, you control which devices NerveCenter can effectively manage. The major SNMP MIB definitions and Open 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 176	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 178	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 181	Describes how to use the MIB compiler to compile the NerveCenter MIB.
Configuring NerveCenter to Use the New MIB on page 182	Describes how to use the NerveCenter Administrator to load the MIB.
Troubleshooting: Managing Management Information Bases (MIBs) on page 184	Lists common problems users face when using MIBs.

Sources of MIB Definitions

Several MIB definitions—the Open 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 176
- *Definitions Not Compiled into the NerveCenter MIB* on page 177
- *Definitions from other sources* on page 178

Definitions Compiled into the NerveCenter MIB

The definitions that are already compiled into the NerveCenter MIB are the Open management application definitions and the major Internet standard SNMP definitions. The definitions are in subdirectories in the mib directory (mibs on UNIX).

Table 13-1 lists the subdirectories that contain these definitions and includes descriptions of each.

Table 13-1. Definitions Compiled into the MIB

This directory...	Includes definitions for...
nsmg	Open 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 13-2. 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
hp	HP-UX servers using the HP-UX SNMP agent
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 223.

- 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 176 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 `nervectr38.mib and mibcomp38.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 recieve 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:

```
interfaces  OBJECT 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
SYNTAX TimeTicks	SYNTAX TimeTicks
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 181.

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.

❖ To compile the MIB:

1. At a command prompt, change to the mib directory (mibs on UNIX).

2. Type the following command. Then press Enter.

   `..\bin\mibcomp mibcomp.txt`

Note Use slashes, instead of backslashes, on UNIX.

Tip You can use the -trace option to monitor the process of compiling your MIB. The syntax is `..\bin\mibcomp -trace mibcomp.txt`.

Caution If you have removed files from mibcomp.txt, you must run mibcomp with the `-clean` argument. For example:
`../bin/mibcomp -clean mibcomp.txt`.
For more information about using mibcomp, see *mibcomp* on page 240.

If you have used previous versions of NerveCenter, please note that you do not need the @.

As the compiler 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. For tips on solving compilation errors, see *Troubleshooting: Managing Management Information Bases (MIBs)* on page 184.

3. If you did not compile the MIB on the machine that is running the NerveCenter Server, copy it to the server machine now.

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.

You have updated the MIB that includes the definitions you need. Now, you must use the NerveCenter Administrator to load this MIB. See, *Configuring NerveCenter to Use the New MIB* on page 182.

Configuring NerveCenter to Use the New MIB

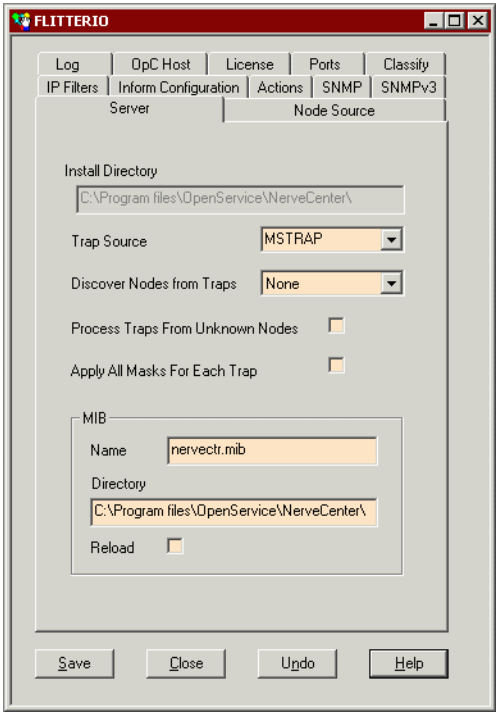
After you have compiled the NerveCenter MIB, you must reload it. From the Administrator, you can specify the name and location if you created a new MIB file and then load the file.

Note You must be a member of NerveCenter Admins on Windows or a member of ncadmins on UNIX.

❖ To load the new MIB:

1. From the Administrator, connect to the server.
2. Select the Server tab.

   NerveCenter displays the Server tab.



3. If the name of the MIB has changed, type the name of the new NerveCenter MIB in the Name field.
4. If the location of the MIB has changed, type the full path of the new MIB in the Directory field.

5. Check the Reload check box.

6. Select Save.

   NerveCenter displays a message that tells you the data was saved successfully. Clients that are connected to this server automatically receive the updated MIB information and display a message to users.

Caution Any clients that are currently connected to this server and have the Share MIB check box selected are updated to use the new MIB, even if they were sharing a MIB from another server.

7. If you want other NerveCenter Servers to use this MIB, repeat this procedure for each NerveCenter Server.

You have specified and loaded the new 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 *Designing and Managing Behavior Models* in the NerveCenter documentation.

Troubleshooting: Managing Management Information Bases (MIBs)

The following list contains some common problems that users encounter when updating the NerveCenter MIBs.

As mentioned in *Adding or Removing MIB Definitions* on page 178, 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 223.

Problem: The mibcomp.txt file is not formatted correctly.

Solution: Check to make sure the appropriate lines are included in the file and are not commented out.

See *Adding or Removing MIB Definitions* on page 178.

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 181.

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 defintion, 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 occurrance 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.

Configuring NerveCenter to Use the New MIB on page 182.

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 181.

Problem: The name or location of the MIB has changed.

Solution: Update the name and location of the MIB and try loading it again.

See *Configuring NerveCenter to Use the New MIB* on page 182.

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:

Table A-1. Sections Included in this Appendix

Section	Description
User Interface Messages on page 188	Explains where error messages appear as well as the different types of error messages.
Error Messages on page 190	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:

- In Windows: Run the Event Viewer and display the Application log. Each error message is listed as a line in the log.
- In UNIX: Read the ASCII 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:11

Table A-2. 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 type of error. The error message numbers are organized as follows:

Table A-3. 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:

Error Messages

The following charts 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 188.

The messages include:

Action Manager Error Messages

Following is a list of Action Manager error messages.

Table A-4. Action Manager Error Messages

Error Number	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-4. Action Manager Error Messages (continued)

Error Number	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-4. Action Manager Error Messages (continued)

Error Number	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.
1515	*value*. The network cannot be reached from this host at this time	Make sure the routers are functioning properly.

Table A-4. Action Manager Error Messages (continued)

Error Number	Error	Resolution
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.
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-5. Alarm Filter Manager Error Messages

Error Number	Error	Resolution
1	Lookup failed on linenumber *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-6. Deserialize Manager Error Messages

Error Number	Error	Resolution
1	Lookup failed on linenumber *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-7. Flatfile Manager Error Messages

Error Number	Error	Resolution
1	Lookup failed on linenumber *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-8. Inform NerveCenter Manager Error Messages

Error Number	Error	Resolution
1	Lookup failed on linenumber *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-9. Inform OV Manager Error Messages

Error Number	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 configuration settings for NetNodeNotify.
1003	No platform host for InformOV.	Use the Administrator to check the configuration settings for NetNodeNotify.
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 configuration settings for NetNodeNotify.
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.

Table A-9. Inform OV Manager Error Messages (continued)

Error Number	Error	Resolution
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.

Table A-9. Inform OV Manager Error Messages (continued)

Error Number	Error	Resolution
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.
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-10. Log to Database Manager Error Messages

Error Number	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-11. Log to File Manager Error Messages

Error Number	Error	Resolution
1	Lookup failed on linenumber *value* in File *value*.	
3001	LogToFile Manager Initialization successfully finished	

OpC Manager Error Messages

Following is a list of OpC Manager error messages.

Table A-12. Inform OpC Manager Error Messages

Error Number	Error	Resolution
1	Lookup failed on linenumber *value* in File *value*.	
3001	OpC Manager Initialization successfully finished	

Poll Manager Error Messages

Following is a list of Poll Manager error messages.

Table A-13. Poll Manager Error Messages

Error Number	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-14. Protocol Manager Error Messages

Error Number	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
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 a value for the 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 value for the SNMP retries.
3003	Invalid value in configuration settings for default SNMP port, using default of 161.	Use the Administrator to enter a value for the default SNMP port number.

PA Resync Manager Error Messages

Following is a list of PA Resync Manager error messages.

Table A-15. PA Resync Manager Error Messages

Error Number	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)	
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.

Table A-15. PA Resync Manager Error Messages (continued)

Error Number	Error	Resolution
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.
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-16. Server Manager Error Messages

Error Number	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
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.

Table A-16. Server Manager Error Messages (continued)

Error Number	Error	Resolution
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 to version 3.5 standards.
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.
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.

Table A-16. Server Manager Error Messages (continued)

Error Number	Error	Resolution
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 the 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 get the license extended.
	License has expired.	Your NerveCenter evaluation license has expired. Contact sales or support to get the license extended.

Table A-16. Server Manager Error Messages (continued)

Error Number	Error	Resolution
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-17. Trap Manager Error Messages

Error Number	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 SNMP service under Windows.
1003	CTrapManagerWnd::OnInvalidSignature - Error in receiving data on NC socket.	Check for consistency in version numbers of trapper and NerveCenter executables.
1004	Expected MSTRAP or OVTRAP in NerveCenter configuration settings.	Reinstall NerveCenter and make sure you choose appropriate platform integration.
2001	MS Trap service threw exception in GetTrap.	Make sure you aren't accidently making SNMP get requests to port 162.

Table A-17. Trap Manager Error Messages (continued)

Error Number	Error	Resolution
2002	Error processing trap data.	Make sure you aren't accidently making SNMP get requests to port 162.
3001	Trap Manager Initialization successfully finished.	
3002	Check Trapper—Trapper process died. restarting Trapper.	

NerveCenter installation Error Messages (UNIX)

Following is a list of NerveCenter installation error messages.

Table A-18. NerveCenter Installation Error Messages (UNIX)

Error	Resolution
Space under *dirname* is INSUFFICIENT to install Open 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 Open 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 Open 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 Open 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 Open 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, asked if you wanted it to stop them, and you said no. You must manually exit from or kill the processes and re-run the installation script.

Table A-18. NerveCenter Installation Error Messages (UNIX) (continued)

Error	Resolution
hostname is not a valid host name	The host that you provided to the script for integration with another application is not a valid host. Check the name of the host (capitalization, spelling, and so on) and try again.
hostname does not have OpenView installed on it.	Before configuring an OpenView host for Open NerveCenter's integration with Open LANAlert or OperationsCenter, OpenView must already be installed on the host. Stop your Open NerveCenter installation and review the prerequisites.
OpenView has not been configured on this system yet.	Before configuring an OpenView host for Open NerveCenter's integration with LANAlert or OperationsCenter, you must have already done the basic OpenView configuration for the host. Rerun the installation script, make sure to answer "Yes" when questioned whether you want to configure OpenView for this host, and then proceed with your integration with other applications.
I don't know how to install on this architecture	Installation is supported for HP-UX and 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.
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 Open 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 Open 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.

Table A-18. NerveCenter Installation Error Messages (UNIX) (continued)

Error	Resolution
Can't create /etc/rc2.d/K94ncservice on Solaris	The script couldn't create the Open NerveCenter rc script /etc/rc2.d/K94ncservice on Solaris or K940ncservice on HP-UX
Can't create /etc/rc2.d/K940ncservice on HP-UX	. Copy *installation_path*/bin/rc.openservice to /etc/rc2.d//K94ncservice on Solaris or K940ncservice on HP-UX . 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-19. OpenView Configuration Error Messages (UNIX)

Error	Resolution
Configuration of OpenView was not entirely successful. You need to go back and double-check the steps that failed above.	This message will be displayed if any part of the OpenView configuration didn't succeed. Scroll back through the output of the script, looking for messages that include the word *FAILED*. Immediately following such a line will be the specific system 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.

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 214	Lists some of the most common problems NerveCenter users face.
Using Trace Counters to Troubleshoot NerveCenter on page 216	Explains how advanced NerveCenter users can use trace counters to detect problems.
Troubleshooting ASN.1 files on page 223	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 33.

An alarm causes the NerveCenter Server to crash every time I start it on page 34.

The NerveCenter Server does not start on Windows on page 34.

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 34

UNIX will not start the NerveCenter Administrator on page 44.

While trying to connect to a NerveCenter Server I get the message: The server did not respond on page 44.

I misspelled a server name while trying to connect and now the misspelled name appears in the NerveCenter Administrator Server Name list on page 44.

When I try to connect to a NerveCenter Server it tells me: Number of allowed client connection exceeded on page 60.

I need to make the same changes to several NerveCenter Servers on page 60.

Importing imputil.ini caused unwanted changes to a NerveCenter Server on page 60.

NerveCenter's node list

NerveCenter is not filtering a node by a capability on page 95.

After setting an IP filter, a node that should be masked out still appears in the node list on page 95.

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 95.

NerveCenter does not recognize my network management platform as a valid source of node data on page 96.

NerveCenter is not receiving node data from my network management platform on page 96.

The IPSweep behavior model will not work on page 96.

The NerveCenter node list contains two nodes with the same address but different names on page 97.

NerveCenter deletes a node I added manually on page 97.

NerveCenter adds a node I deleted manually on page 97.

I'm seeing several errors recorded in the application event log window stating that ipsweep.exe is not running on page 97.

Alarm actions

NerveCenter is not sending SMTP mail notifications on page 151.

Microsoft mail notifications are not being sent as NerveCenter on page 151.

NerveCenter is not dialing pages correctly on page 151.

The logs created by my behavior models are using too much disk space on page 151.

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 174

Running SerializeDB from the command line keeps failing on page 174.

SerializeDB won't work from the application or the command line on page 174.

The data in the database is not what was imported from the serialized file on page 174.

.

Management Information Bases (MIBs)

A MIB won't compile on page 184.

A MIB won't load on page 186.

Using Trace Counters to Troubleshoot NerveCenter

A useful tool for troubleshooting NerveCenter is it 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; or
- If you have reset a counter, since the reset was performed.

If you are having difficulties with NerveCenter, a Open 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. The following table explains what procedures are related to the trace counters:

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**.

*. Each of these procedures are performed within a NerveCenter Administrator after it is connected to a NerveCenter Server.

The following sections describe the meaning of each counter:

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 216 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 216 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 216 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.
OPC	Requested	The number of informs requested to be sent to NerveCenter's SEMSOPCA adapter.
	Sent	The number of informs that have been sent to NerveCenter's SEMSOPCA 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 216 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 216 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 216 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.

Table B-7. Log to File Counters (continued)

Field	Description
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 216 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 216 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.

OPC Msg Counters

The OPC Msg 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 216 for more information about using the counters.

Table B-10. OPC Msg Counters

Field	Description
Msgs received	The number of messages that the current NerveCenter Server has received from HP IT/Operations.
Msgs processed	The number of HP IT/Operations messages that the current NerveCenter Server has processed through an OpC mask.

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 216 for more information about using the counters.

Table B-11. 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.

Table B-11. Poll Counters (continued)

Group	Field	Description
	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 216 for more information about using the counters.

Table B-12. 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 216 for more information about using the counters.

Table B-13. Trigger Counters

Field	Description
Masks	The number of triggers that NerveCenter has fired from trap masks.
OpC Masks	The number of triggers that NerveCenter has fired from OpC masks.
Actions	The number of triggers that NerveCenter has fired from alarm actions.

Table B-13. Trigger Counters (continued)

Field	Description
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-14. 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.

Controlling NerveCenter from the Command Line C

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 UNIX Shell/DOS Prompt Commands on page 230	Describes the NerveCenter commands that you can execute from a UNIX shell or a Windows command prompt.
NerveCenter Command Line Interface on page 270	Explains the NerveCenter command line interface (CLI) and how to use it.

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:

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 the manual, *Monitoring Your Network*.

USAGE

On UNIX, if you receive the error message `client: Command not found`, NerveCenter has not been installed in the default location (`/opt/OSInc`). In this case, you must change directories to the NerveCenter bin directory before entering the command shown above, or enter the full pathname of the executable.

Note On UNIX, before you can run NerveCenter, you must first set the necessary UNIX environment variables by running the appropriate ncenv shell script. For more information about setting environment variables, refer to the book *Managing NerveCenter*.

see also

ncstop on page 255

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. You can use either one to set up your database. If you are installing NerveCenter on many computers, you may want to use the wizard to set up the first database, then 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 235

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 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.

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 the section **[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, *Information About imputil.ini Sections* on page 234.

Table C-1. Information About imputil.ini Sections

For Information About...	Refer to This Book...
CONFIG_PLATFORM	Integrating NerveCenter
CONFIG_PLATFORM_MAPSUBNETS	Integrating NerveCenter
CONFIG_PLATFORM_NETNODENOTIFY	Integrating NerveCenter
CONFIG_SERVER	Managing NerveCenter
CONFIG_SERVER_PATH	Managing NerveCenter
CONFIG_SERVER_PAGER	Managing NerveCenter
CONFIG_SERVER_SNMP	Managing NerveCenter
CONFIG_SERVER_SNMPV3	Managing NerveCenter
CONFIG_SERVER_LOGS	Managing NerveCenter
CONFIG_SERVER_MSMAIL	Managing NerveCenter
CONFIG_SERVER_SMTPMAIL	Managing NerveCenter
IMPORT_MODEL	Designing and Managing Behavior Models
IMPORT_NODE	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

All command line switches are case-insensitive (Windows only) and 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 a dialog box with 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 SNMP v1 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 *Managing NerveCenter* and *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 locally on the machine hosting NerveCenter Server. Configure security on the server machine appropriately so no unauthorized users can overwrite or change the MIB.

After compiling the NerveCenter MIB, you must reload it. Refer to the manual *Managing NerveCenter* for more information.

ncadmin

NAME

ncadmin

SYNOPSIS

```
ncadmin
```

DESCRIPTION

Executes the NerveCenter Administrator application. For information about how to use NerveCenter Administrator, refer to the manual, *Managing NerveCenter*.

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 32.

see also

ncserver on page 251, *ncstop* on page 255

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 name or path name that contains spaces, the name must be enclosed in quotation marks, for example, “C:\Program Files\OpenService\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. Once you've performed this step, you can type **`nccmd -f`** *scriptfile* from a Windows command prompt or a UNIX shell 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 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 244.

`-?`

Provides command-line help for the nccmd command.

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.

This reference consists of the following sections:

- *Server commands* on page 244
- *Object commands* on page 247
- *General commands* on page 249
- *Notes for CLI commands* on page 250
- *CLI command examples* on page 250

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 earlier.

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.

`clear trigger -t` *triggerName* `[-i` *alarmInstanceID*`]`

Cancels any pending triggers of a specified name that have been queued by its own alarm instance. If no alarm instance is supplied, then NerveCenter cancels the specified trigger for all alarms in which the trigger is pending.

-t *triggerName*

Name of the trigger to be cleared.

`-i` *alarmInstanceID*

(Optional) *alarmInstanceID* is a numeric value which indicates the instance of the alarm 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 a numeric value (in seconds) for NerveCenter to wait before it fires the specified trigger.

`list`

Lists connected servers.

`select -s` *serverName*

Makes *serverName* the active server. *serverName* -is one of the connected NerveCenter Servers.

`set server {-t` *timeoutInterval*`} | {-w` *width*`} | -?`

timeoutInterval is the number of seconds that the CLI waits for an acknowledgement from the NerveCenter Server that a CLI request was received. The default value for timeouts 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, connection to the Server is still maintained, but the CLI displays the message: "Error. No data was received within allowed time."

width is the character width (1-50) of columns for tabular output that the NerveCenter CLI **`list`** command displays. The default for *width* is 12 characters.

-? displays help for the **`set server`** commands.

`-`*switch parameter*

One of several switch/parameter pairs described below. Parameter switches can be preceded by either a hyphen (-) or a backslash (\).

`show server -`*serverSwitch*

Displays information about the active server.

`-`*serverSwitch*

One of the several server-related switches described below:

`-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 the specified *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.

`{delete | list` *object* `{-`*switch parameter*`} | -? } |`
`{set` *object* `{-`*switch parameter* `-w` *targetObject*`} | -? }`

`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, displays a list of 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.

`-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 SNMP v3 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 SNMP v3 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`**.

`-s enterprise | node | subobject | instance`

Identifies NerveCenter alarms by their scope.

`-s yes | no`

Identifies NerveCenter nodes their state: suppressed (yes) or unsuppressed (no).

`-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

`-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 the NerveCenter objects whose attributes are being set.

`-x v3`

Displays extended attributes of all SNMP v3 nodes. These attributes (security level, authentication protocol, and error status) are not displayed by the **`list node -n`** command.

`-?`

Display help information about the delete/list/set *object* command.

`set`

Sets attributes for the 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. If used without an action keyword or *object*, displays general information about help. If used with an action alone, gives general information about the command, including the command's 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 connected 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
```

ncserver

NAME

ncserver

SYNOPSIS

```
ncserver [-fg | -info | -off]
```

DESCRIPTION

Executes the NerveCenter Server application. For information about how to use NerveCenter Server, refer to the manual, *Managing NerveCenter*.

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.)

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 253

ncstart

NAME

ncstart

SYNOPSIS

```
ncstart -h | -p | -s
```

DESCRIPTION

Runs the NerveCenter Server as either a process or a Windows service.

OPTIONS

Note These options are only available on Windows machines. These options do not function on UNIX.

`-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 These options are only available on UNIX machines. These options do not function on Windows.

`-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.)

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 251, *ncstop* on page 255

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 231, *ncadmin* on page 242, *ncserver* on page 251, *ncstart* on page 253

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 ]
```

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 NerveCenter with a Network Management Platform.*

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 OpenVien 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.

`-info`

print out 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 the network management platforms Micromuse Netcool/OMNIbus and CA Unicenter TNG.

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 Micromuse 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 Micromuse 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`**.

`-t`

Enables or disables NerveCenter integration with Tivoli TME.

`-tcfg`

Defines the full qualified path/filename for the Event Adapter configuration file. The default is `/opt/OSInc/userfiles/nctec.cfg`

`-thost`

Defines the machine on which the Tivoli Event Server is located. The default is localhost.

`-toneway`

Defines the best-effort delivery method to the Tivoli TME agent. The default is 0.

`-tport`

Defines the port the NerveCenter platform adapter uses to communicate with Tivoli Event Server. The default is 32510.

`-ttype`

Defines 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.

`-u`

Enables or disables NerveCenter integration with CA Unicenter TNG.

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`**.

`-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`**.

semsopca

NAME

semsopca

SYNOPSIS

```
semsopca [-agent] [-v1]
```

DESCRIPTION

A Open-supplied interface that enables NerveCenter to send and receive Hewlett Packard's OpenView IT/Operations (ITO) messages. NerveCenter setup installs the IT/O Platform Adapter when OVPA platform integration is selected.

OPTIONS

`-agent`

Connects to the OpC agent interface. By default, `semsopca` connects to the OpC server interface.

`-v1`

Pipes `semsopca` messages to the UNIX shell console.

USAGE

The **`-v1`** switch is equivalent to setting the UNIX environment variable **`NL_OPC_VERBOSE = 1`**.

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 a 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

Generates an SNMP-style trap or inform for NerveCenter testing purposes.

OPTIONS

`-v1`

Sends a SNMP v1 trap. This is the default

`-v2c`

Sends a SNMP v2c trap.

`-v2cinform`

Sends a SNMP v2c inform.

`-v3inform`

Sends a SNMP v3 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, you should 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. If you are sending a v3inform, you must supply a privacy protocol.

`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 the following: integer, octetstring, octetstringhex, octetstringoctal, octetstringascii, objectidentifier, null, ipaddress, counter, gauge, timeticks, opaque, opaquehex, opaqueoctal, or opaqueascii, counter 64 (v2c, v2cinform, and v3inform only), uint32 (v2c, v2cinform, and v3inform only).

value

Value of *variable*. Enclosing strings with quotes is optional.

examples

Example 1: In the following example, Trapgen sends an generic SNMP v1 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 1.3.6.1.4.1.9.1.3 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
```

NerveCenter Command Line Interface

You can use a command line interface (in a UNIX shell or a DOS Command Prompt window) to list, delete, and control nodes, alarms, polls, and masks. You can also connect to, display the status of, and disconnect from NerveCenter servers. You can issue the commands manually or from a script.

The following sections describe how to control NerveCenter from the command line using one of the following methods:

- Entering a single command in a shell or a DOS prompt
- Entering multiple commands in an interactive session
- Running a script that contains one or more commands

For information about specific commands, see the section *Command line interface reference* on page 244. The procedures in this appendix assume 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.

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, follow these steps:

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 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 *Command* is the command you want to run. Then press Return.

Note The user ID and password are encrypted before they are 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 syntax for the commands is specified in *Command line interface reference* on page 244.

The command is executed and informs you if an error was encountered.

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, follow these steps:

1. Open a UNIX shell or a DOS Command Prompt window.

2. Do one of the following:

To open an interactive session without connecting directly to a server, type **nccmd** and press Return.

The nccmd command prompt is displayed:

```
nc>
```

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 user name and password for that NerveCenter server, and *Port* is the server's command-line-interface port (32506 by default). Then press Return.

The nccmd command prompt is displayed and you are connected to the server you specified:

```
nc>
```

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 Return after each one.

   The syntax for the commands is specified in *Command line interface reference* on page 244.

   Each time you press Return, 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 Return.

   You are returned to a shell prompt or a 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 that contains one or more commands

❖ **To run a script that contains one or more commands, follow these steps:**

1. Create a text file that contains the commands in the order you would enter them if you were in an interactive session.

   The syntax for the commands is specified in *Command line interface reference* on page 244.

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\Open Software\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
```

Index

K

L

M

N