Open NerveCenter

Monitoring Your Network

Windows and UNIX

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

Trademarks

Printed in the USA, v4.0.
Open NerveCenter *Monitoring Your Network*
OpenService, Inc.
110 Turnpike Road, Suite 308
Westborough, MA 01581
Phone 508-366-0804
Fax 508-366-0814
http://www.openservice.com

Contents

C

1 Introduction

2 Understanding NerveCenter

5 Monitoring Alarms

6 Resetting Alarms

A Communications and Data

B Error Messages

Contents

Introduction

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

This chapter includes the following sections:

Section	Description
Overview of this Book on page 2	Includes an overview of the contents of this guide and what you need to know before you use the guide.
NerveCenter Documentation on page 3	Lists and describes the components of the Open NerveCenter support system, including printed guides, online guides, help, and links to the Open NerveCenter Web site and the Open technical support Web site.
Open Technical Support on page 7	Describes how to access the NerveCenter knowledge base and other Open support services.

1 Introduction

Overview of this Book

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

Monitoring Your Network contains the following chapters:

Title	Description
Chapter 2, *Understanding NerveCenter*	Discusses what NerveCenter is and how it works within your overall network management strategy.
Chapter 3, *Getting Started with NerveCenter Web Client*	Explains how to perform basic NerveCenter Web Client tasks such as: starting NerveCenter and connecting to a NerveCenter server; setting preferences; and disconnecting from a server.
Chapter 5, *Monitoring Alarms*	Discusses how to interpret the information provided by the NerveCenter alarm monitoring interfaces and how to examine an alarm instance’s history.
Chapter 4, *Getting Started with NerveCenter Client*	Explains how to perform basic NerveCenter tasks such as: starting NerveCenter and connecting to a NerveCenter server; defining alarm-instance filters; and disconnecting from a server.
Chapter 6, *Resetting Alarms*	Documents the ways the NerveCenter Web Client and the NerveCenter Client enable you to reset alarm instances.
Chapter 7, *Monitoring SNMP Status and Operations*	Describes SNMP v3 error status and the operations log.
Chapter 8, *Monitoring Nodes*	Explains how the NerveCenter Web Client and the NerveCenter Client enable you to monitor and obtain information about network nodes.
Chapter 9, *Generating Reports*	Describes how to add and generate reports in the Windows environment.
Chapter 10, *Checking the Status of the Server*	Documents the various pages of the Server Status dialog that you use to obtain information about the active NerveCenter server.
Appendix A, *Communications and Data*	Discusses how NerveCenter communicates with other processes.
Appendix B, *Error Messages*	Lists the error messages that exist in NerveCenter.

NerveCenter Documentation

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

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

- *Using the Online Help* on page 3
- *Printing the Documentation* on page 3
- *The NerveCenter Documentation Library* on page 4
- *UNIX Systems* on page 5
- *Document Conventions* on page 5
- *Documentation Feedback* on page 6

Using the Online Help

You can use Microsoft Internet Explorer, Mozilla, or Netscape Navigator to view the documentation. Refer to the *NerveCenter 4.0.3 Release Notes* for the browser versions supported with this release.

TIP

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

Printing the Documentation

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

NOTE

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

The NerveCenter Documentation Library

The following documents ship with NerveCenter.

Book Title	Description	Application	Audience	PDF for Print
NerveCenter 4.0.3 Release Notes	Describes new NerveCenter v4.0 features and includes late-breaking information, software support, corrections, and instructions.	All	All	relnotes.pdf
Getting a Quick Start with NerveCenter	Provides a preview prior to installing and configuring NerveCenter for your network. Takes approximately one to two hours to complete.	All	New users	QuickStart.pdf
Upgrading NerveCenter	Explains how to upgrade your current NerveCenter version.	All	Installation team	upgrading.pdf
Installing NerveCenter	Helps you plan and carry out your NerveCenter installation. Use the *Release Notes* in conjunction with this book.	All	Installation team	install.pdf
Managing NerveCenter	Explains how to customize and tune NerveCenter after it has been installed.	NerveCenter Administrator	Administrator	managing_ nervecenter.pdf
Integrating NerveCenter with a Network Management Platform	Explains how to integrate NerveCenter with network management platforms.	NerveCenter Administrator	Administrator	integratingNC. pdf
Learning How to Create Behavior Models	Provides step-by-step instructions and examples for creating behavior models.	NerveCenter Client	Users with administrative privileges	learningModel. pdf
Designing and Managing Behavior Models	Explains behavior models in depth, how to create or modify models, and how to manage your models.	NerveCenter Client	Users with administrative privileges	designingModels. pdf
Monitoring Your Network	Explains how NerveCenter works and how you can most effectively monitor your network.	NerveCenter Client and Web Client	Users	monitoringNet. pdf
Behavior Models Cookbook	Describes each behavior model shipped with Open NerveCenter.	NerveCenter Client	Users with administrative privileges	modsCookbook. pdf

Book Title	Description	Application	Audience	PDF for Print
Quick reference cards	Quick reference cards provide convenient reference material for common NerveCenter tasks. The following cards are provided (PDF only): ◆ Monitoring NerveCenter reference. ◆ Installing NerveCenter reference. ◆ Using Behavior Models reference.	NerveCenter Client and Administrator	All	quickreference. pdf

UNIX Systems

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

Document Conventions

This document uses the following typographical conventions:

Element	Convention	Example
Key names, button names, menu names, command names, and user entries	**Bold**	Press **Tab** Enter **ovpa -pc**
◆ A variable you substitute with a specific entry ◆ Emphasis ◆ Heading or Publication Title	*Italic*	Enter **./installdb -f** *IDBfile*
Code samples, code to enter, or application output	`Code`	`iifInOctets > 0`
Messages in application dialog boxes	Message	Are you sure you want to delete?
An arrow (>) indicates a menu selection	>	Choose **Start** > **Programs** > **Open NerveCenter**

Element	Convention	Example
A link to a section in the same book	*Blue Italic*	For more information, see *Correlating Conditions*.
A link to a section in a different book **Note:** If you are viewing this document in a PDF viewer, you may need to use the **Go to Previous View** button to return to the original PDF file.	*Green Italic*	For more information, see *Correlating Conditions* in *Monitoring Your Network with NerveCenter*.

CAUTION

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

NOTE

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

TIP

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

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

Documentation Feedback

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

Documentation
OpenService, Inc.
110 Turnpike Road, Suite 308
Westborough, MA 01581

documentation@openservice.com

Open Technical Support

Open is committed to offering the industry's best technical support to our customers and partners. You can quickly and easily obtain support for NerveCenter, our proactive network management software, or Security Threat Manager, our security threat management suite.

Professional Services

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

Educational Services

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

Contacting the Customer Support Center

For telephone support:

Phone: 1-888-886-1085, menu option 1 or 1-508-599-2000

For e-mail support:

E-mail: techsupport@openservice.com.

For electronic support

Open uses a Web-based customer call tracking system, TeamShare, where you can enter questions, log issues, track the status of logged incidents, and check the knowledge base.

When you purchased your product or renewed your maintenance contract, you received a user name and password to access the TeamShare system. If you have not received or have forgotten your log-in credentials, please e-mail us with a contact name and company specifics at techsupport@openservice.com.

For Online KnowledgeBase Access

For additional NerveCenter support, you can access the following information on the Open website, http://www.openservice.com:

- **Patches and Updates** - latest installation files, patches, and updates including documentation for NerveCenter.
- **Software Alerts** - latest software alerts relative to NerveCenter.
- **KnowledgeBase Search** - search the NerveCenter KnowledgeBase for answers to your questions whether relating to installation, usage, or operation.

Understanding NerveCenter

This chapter explains:

- What type of product NerveCenter[TM] 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 2-1. Sections Included in this Chapter

Section	Description
What is NerveCenter? on page 10	Explains that NerveCenter is an advanced event automation solution.
How NerveCenter Manages Nodes on page 11	Explains how NerveCenter isolates and responds to emerging network and system problems.
Main NerveCenter Components on page 21	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 29	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 11
- *Detecting Conditions* on page 12
- *Correlating Conditions* on page 12
- *Responding to Conditions* on page 17

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 29, 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 13
- *Finding a Set of Conditions* on page 14
- *Looking for a Sequence of Conditions* on page 15

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.

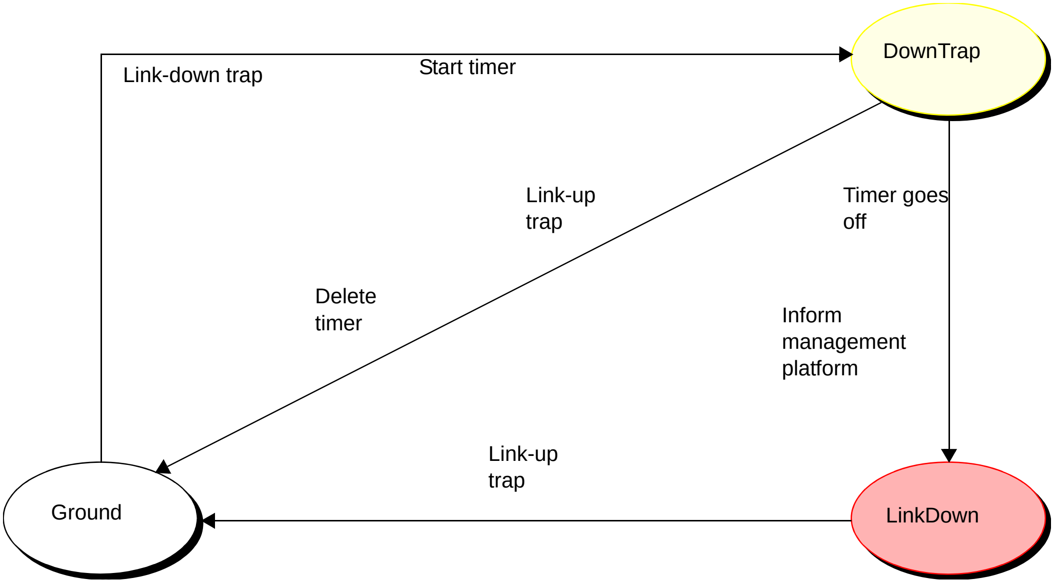


FIGURE 2-1. State Diagram for Detecting a Link-Down Condition

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.

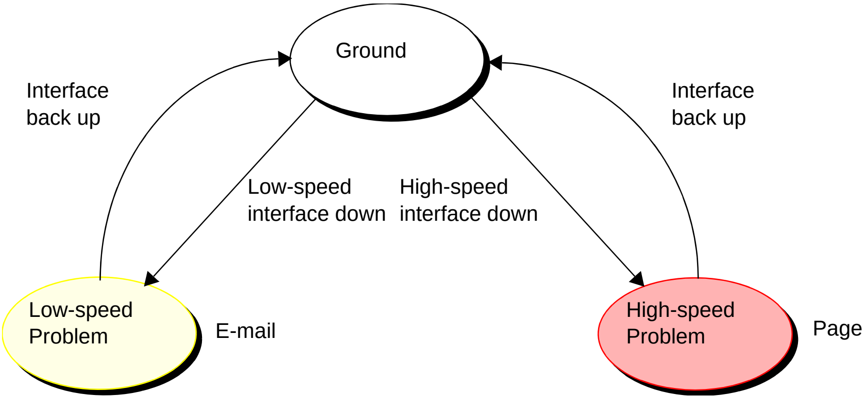


FIGURE 2-2. State Diagram for Detecting a Router Interface Problem

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.

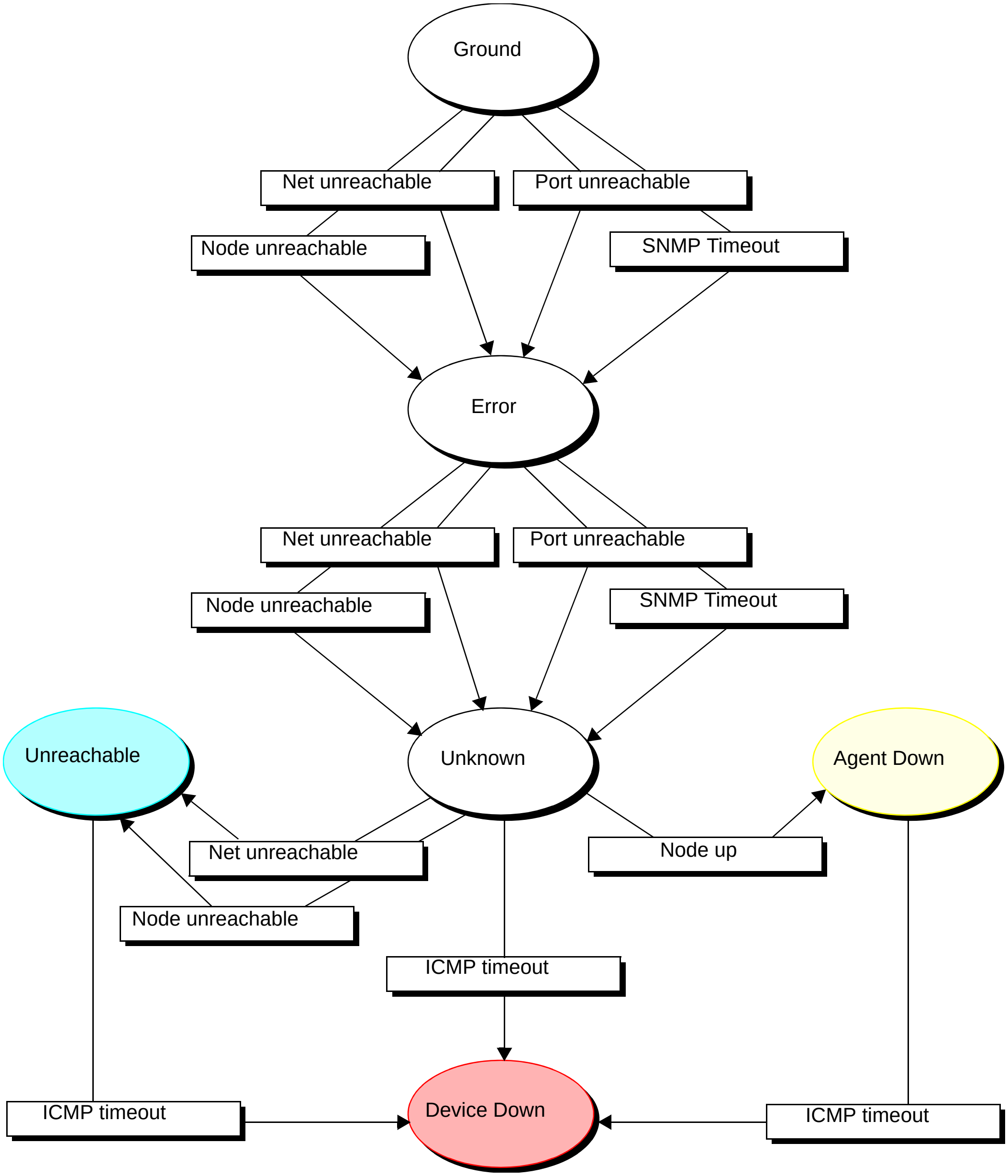


FIGURE 2-3. State Diagram for Determining Node Status

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 18
- *Logging* on page 18
- *Causing State Transitions* on page 19
- *Corrective Actions* on page 19
- *Action Router* on page 20

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 29.

Logging

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

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

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

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

Causing State Transitions

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

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

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

NerveCenter also includes a Send Trap action. You define the trap to be sent, including the variable bindings, and associate the action with a state transition. When the transition occurs, the trap is sent. The trap can be caught by a NerveCenter trap mask—in which case you can use Send Trap somewhat like Fire Trigger, to generate a trigger—or by any application 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 21
- *The NerveCenter Database* on page 22
- *The NerveCenter User Interface* on page 25

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 22.

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 24.

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 2-4 tracks the level of traffic on an interface.

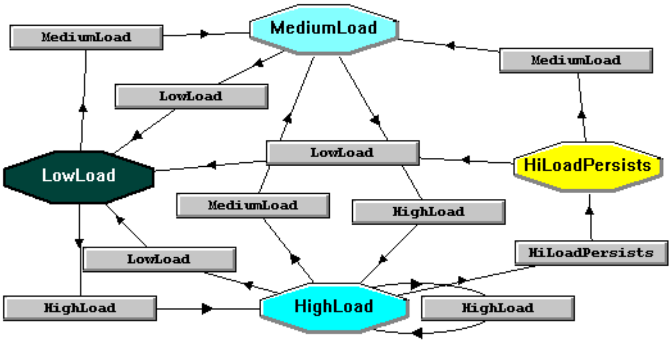


FIGURE 2-4. Alarm State Diagram

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 *Behavior Models and Their Components* in *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

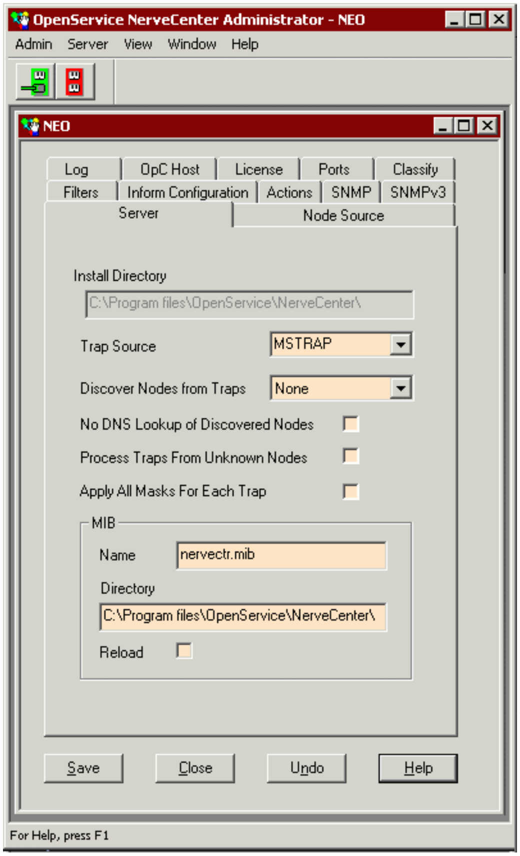
The NerveCenter Administrator is used to configure NerveCenter once it has been installed. The NerveCenter Client and the NerveCenter Web Client are used to monitor a network for problems. The NerveCenter Client is also used to create new behavior models. The command line interface can be used to perform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

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

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

The NerveCenter Administrator

Figure 2-5 shows the graphical user interface (GUI) for the NerveCenter Administrator.



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

Trap Source: MSTRAP

Discover Nodes from Traps: None

No DNS Lookup of Discovered Nodes ☐

Process Traps From Unknown Nodes ☐

Apply All Masks For Each Trap ☐

MIB

Name: nervectr.mib

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

Reload ☐

FIGURE 2-5. 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.

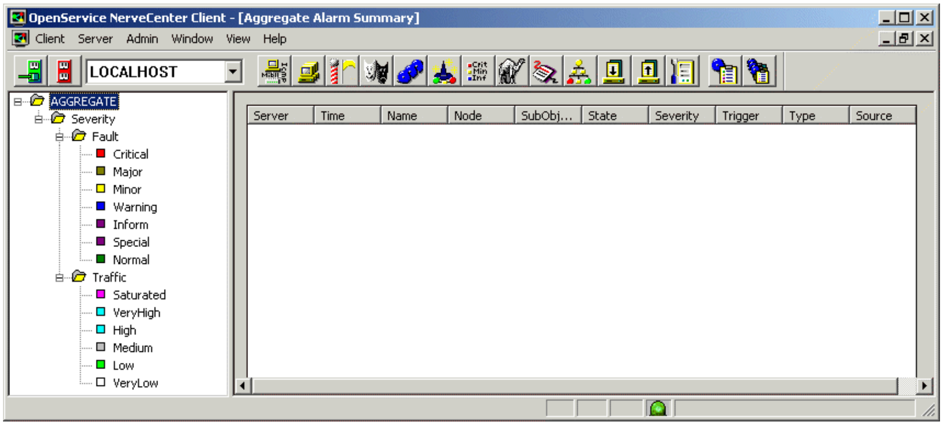


FIGURE 2-6. NerveCenter Client

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

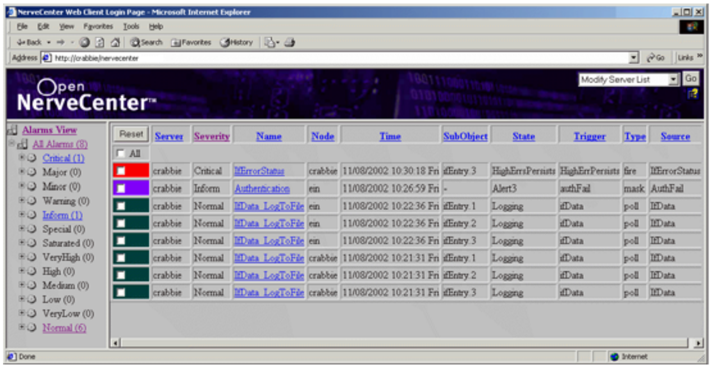
For complete information on using the NerveCenter Client to perform the tasks listed above and others, see *Monitoring Your Network*.

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.



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

FIGURE 2-7. 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 *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.

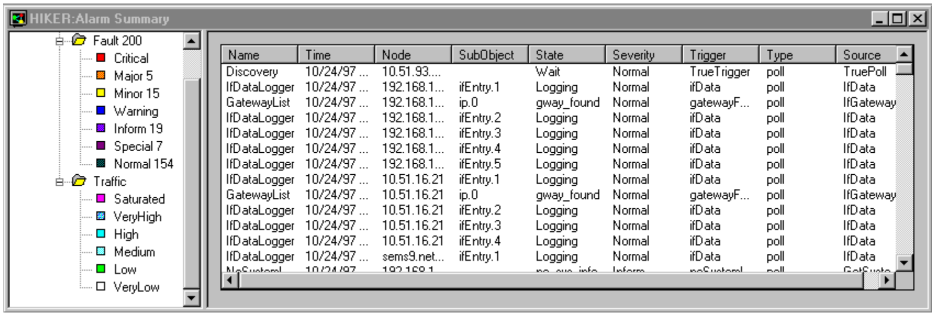
Role in Network Management Strategy

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 29 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 30 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 32 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 33 for further information.

Standalone Operation

At smaller sites, you can use NerveCenter alone for your network management tasks. As we've seen, NerveCenter is very strong in the areas of event correlation and automated actions. In addition, NerveCenter includes an alarm console, as shown in Figure 2-8.



- Fault 200
  - 🟥 Critical
  - 🟧 Major 5
  - 🟨 Minor 15
  - 🟦 Warning
  - 🟪 Inform 19
  - 🟪 Special 7
  - ■ Normal 154
- Traffic
  - 🟪 Saturated
  - ▨ VeryHigh
  - 🟦 High
  - ▨ Medium
  - 🟩 Low
  - □ VeryLow

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

FIGURE 2-8. 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.

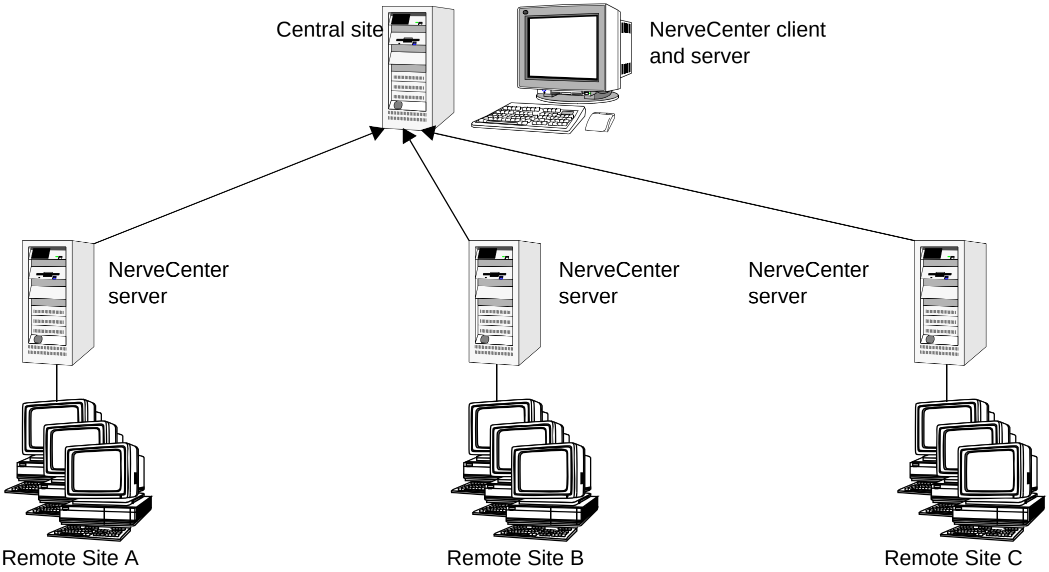


FIGURE 2-9. Distributed NerveCenter Servers

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 section, *Integration with NMPs for Node Information* on page 33, 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

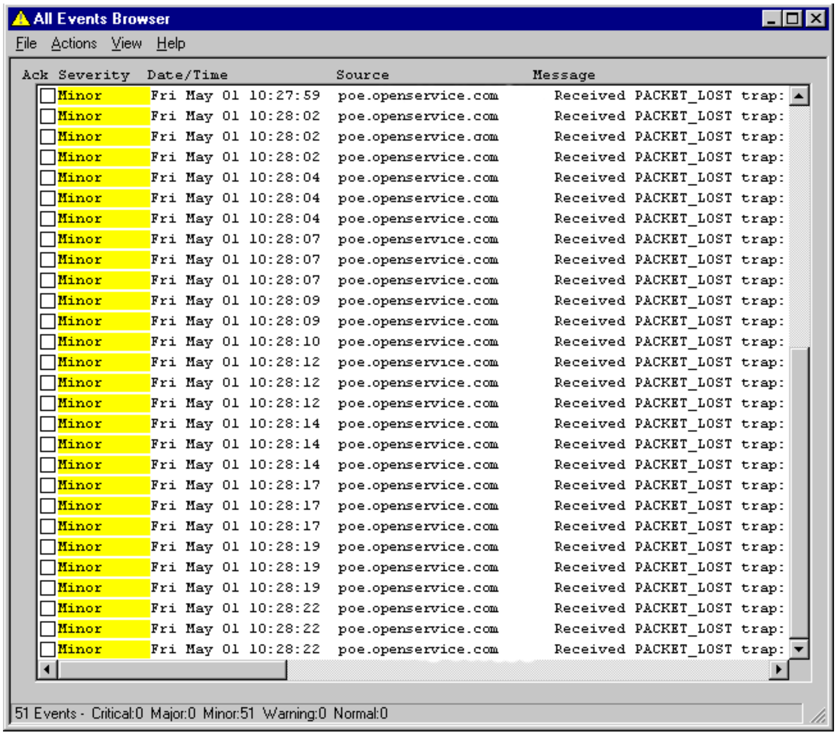
Integration with NMPs for Node Information

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

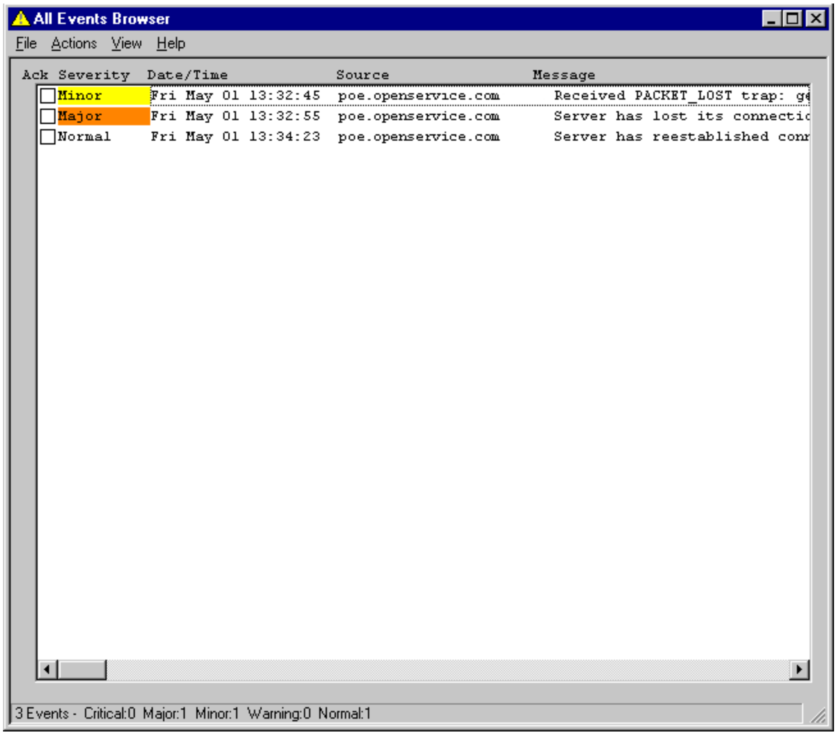
Figure 2-10 below shows an OpenView event browser that contains a flurry of events all caused by the same problem. Figure 2-11 shows what might appear in the browser if NerveCenter were used to screen and correlate the conditions and pass on only important information to the platform event browser.



Ack	Severity	Date/Time	Source	Message
□	Minor	Fri May 01 10:27:59	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:02	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:02	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:02	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:04	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:04	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:04	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:07	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:07	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:07	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:09	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:09	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:10	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:12	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:12	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:12	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:14	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:14	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:14	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:17	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:17	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:17	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:19	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:19	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:19	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:22	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:22	poe.openservice.com	Received PACKET_LOST trap:
□	Minor	Fri May 01 10:28:22	poe.openservice.com	Received PACKET_LOST trap:

51 Events - Critical:0 Major:0 Minor:51 Warning:0 Normal:0

FIGURE 2-10. Too Many Events



```
Ack Severity  Date/Time               Source                 Message
□   Minor     Fri May 01 13:32:45  poe.openservice.com     Received PACKET_LOST trap: ge
□   Major     Fri May 01 13:32:55  poe.openservice.com     Server has lost its connectic
□   Normal    Fri May 01 13:34:23  poe.openservice.com     Server has reestablished conn
```

3 Events - Critical:0 Major:1 Minor:1 Warning:0 Normal:1

FIGURE 2-11. 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.

Getting Started with NerveCenter Web Client

3

This chapter covers the tasks you must perform before you begin monitoring your network. These tasks include starting the NerveCenter Web Client, connecting to a NerveCenter Server, modifying the server connection list, setting preferences for what types of alarm instances you want to monitor, and disconnecting from the server.

For details about how to set up the NerveCenter Web Client, see *Managing NerveCenter Web Integration* in *Managing NerveCenter*.

For explanations of how to perform these tasks, see the following sections:

Section	Description
Starting the Web Client on page 38	Describes how to start the NerveCenter Web Client and log on to one or more NerveCenter Servers.
Modifying the Server Connection List on page 40	Explains how to modify the server connection list which the Web client uses to connect to one or more servers.
Setting Preferences on page 42	Provides instructions for setting up alarm viewing preferences. You can request that the alarm instances from the servers you're connected to be filtered by: server, severity, property group, and partition.
Disconnecting from a Server on page 48	Describes how to log off the NerveCenter Server.

Starting the Web Client

When you start the NerveCenter Web Client, you are prompted for a username and password which the client uses to connect you automatically to one or more NerveCenter servers that you've preselected.

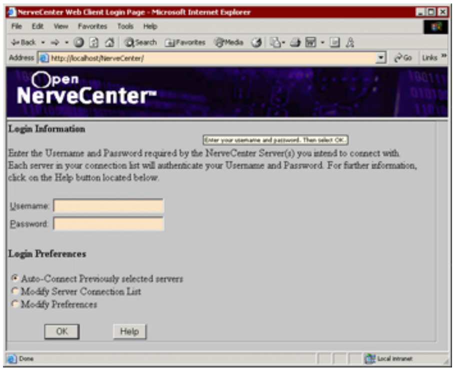
Of course the first time you use the client, no servers will have been selected, so you must specify one or a set of NerveCenter servers that the Web client connects to. See the section *Modifying the Server Connection List* on page 40, for more information.

Once you start the Web client, you can also modify which alarm instances display and how they display in the Web client. See the section *Defining a Partition* on page 47, for more information.

To Start the Web Client

1. Start your Web browser.
2. In your browser's address or location field, enter the URL of the server on which you've installed NerveCenter Web support, followed by a slash, followed by the directory name NerveCenter. For example, you might enter: **http://*durncweb*/NerveCenter** where *durncweb* is the Web server.

   The NerveCenter Web Client's Login Information page displays.

   
   NerveCenter Web Client Login Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address
Go
Links
Open NerveCenter™

   http://localhost/NerveCenter/

   Login Information

   Enter your username and password. Then select OK.

   Enter the Username and Password required by the NerveCenter Server(s) you intend to connect with. Each server in your connection list will authenticate your Username and Password. For further information, click on the Help button located below.

   Username:
   Password:

   Login Preferences

   ● Auto-Connect Previously selected servers
   ○ Modify Server Connection List
   ○ Modify Preferences

3. Enter a user name and password in the **Username** and **Password** fields.

   The user whose name you enter here must be a member of the NerveCenter Users or NerveCenter Admins group (Windows) or the ncusers or ncadmins group (UNIX) on the servers to which you want to connect.
4. Select one of the **Login Preferences** radio buttons.

NOTE

If you’re logging in to a NerveCenter Server using the Web client for the first time, skip this step and go to step 5. Then, to complete your login, see the section, *Modifying the Server Connection List* on page 40, for more information.

- If you select the **Auto-Connect Previously Selected Servers** radio button (the default) after you log in, you will be taken to the client’s alarm-summary page. This is the page you use to view information about alarm instances.
- If you select the **Modify Server Connection List** radio button after you log in, you will be taken to the client’s Server Selection page. This is the page you use to specify the servers from which you want to get information about alarm instances. See the section *Modifying the Server Connection List* on page 40, for more information.
- If you select the **Modify Preferences** radio button after you log in, you will be taken to the client’s Preferences page. This is the page you use to specify how the client should filter and present information about alarm instances. See the section *Setting Preferences* on page 42, for more information.

5. Select the **OK** button.

Modifying the Server Connection List

When you connect to a NerveCenter Server, the Web client automatically connects you to one or more NerveCenter servers that you've preselected. This list of one or more servers is called a *server connection list*.

You modify the server connection list from the Web client's Server Selection page.

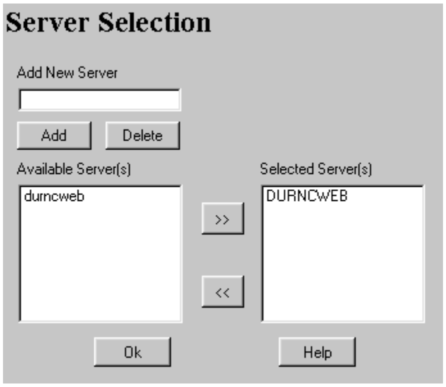
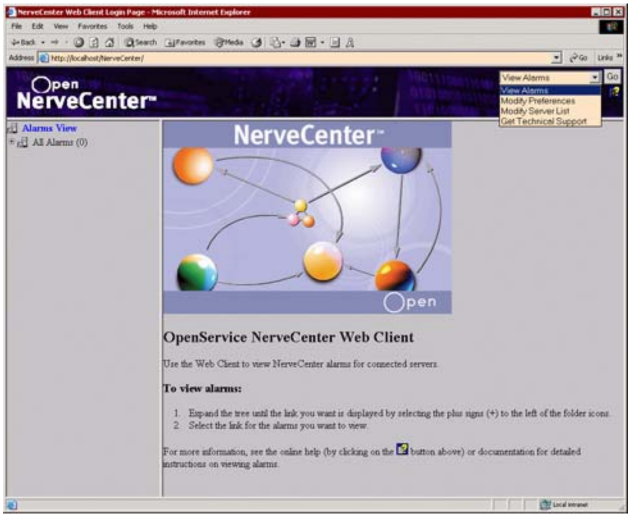


FIGURE 3-1. Server Selection page

The first time you use the Web client, no servers will have been selected, so you must initialize this set of servers. After you've initially specified one or more servers to connect to, you can always go back and modify your server connection list.

To define the set of servers you want to connect to at login

1. Access the Server Selection page. If you are:
   - Starting the Web client for the first time, the Server Selection page is displayed automatically once you select **OK** on the Web client's Login Information page.
   - Starting the Web client any time *after* the first time, in the Login Information page, select **Modify Server Connect List** before selecting **OK**.
   - Already logged in to a NerveCenter Server, in the alarm-summary page, select the **Modify Server List** from the drop-down list box in the upper right corner of the client window, and select the **Go** button.


NerveCenter Web Client Login Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/NerveCenter/
Go Links
Open NerveCenter
View Alarms
Go
View Alarms
Modify Preferences
Modify Server List
Get Technical Support
Alarms View
All Alarms (0)
NerveCenter
Open
Local intranet

OpenService NerveCenter Web Client

Use the Web Client to view NerveCenter alarms for connected servers.

To view alarms:

1. Expand the tree until the link you want is displayed by selecting the plus signs (+) to the left of the folder icons.
2. Select the link for the alarms you want to view.

For more information, see the online help (by clicking on the button above) or documentation for detailed instructions on viewing alarms.

2. In the Server Selection page, populate the **Available Servers** list with the names of all the NerveCenter servers to which you might potentially connect. For each server:

   a. Type the name of the server in the **Add New Server** text field.

   b. Select the **Add** button.

   The **Available Servers** list box now includes the name of the server.

NOTE

The servers that you select must allow logins using the username and password that you supplied when you started the Web client; also, the username must be a member of a NerveCenter user group (unless you are running Windows *without* NerveCenter security). You must use the same username and password for every server to which you want to connect.

3. Move the names of the servers you want to connect to now to the **Selected Servers** list box. For each server:

   a. Select the server in the **Available Servers** list.

   b. Select the >> button.

4. Select the **OK** button.

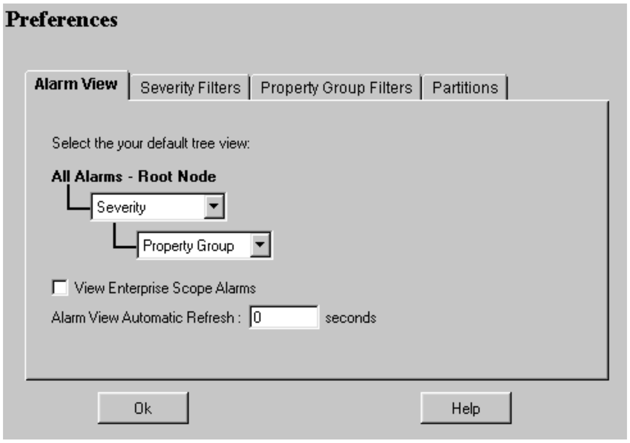
Setting Preferences

Exactly what alarm instances you see on the alarm-summary page and how those instances are presented in the Web client’s tree view depend not only on the server to which you’re connected, but on a set of preferences you set on the Preferences page. This section explains the procedure for setting preferences and discusses how your settings affect what you see in the alarm-summary window.

TO SET YOUR PREFERENCES

1. Go to the Preferences page.

   If you select the **Modify Preferences** radio button while you’re logging on to the client, you’ll be taken to this page. You can also reach this page from the alarm-summary window: just select **Modify Preferences** from the drop-down list box in the upper right corner of the window and select the **Go** button.



Select the your default tree view:

All Alarms - Root Node

Severity

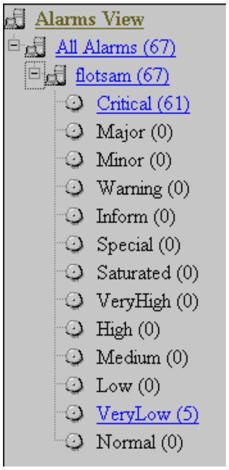
Property Group

□ View Enterprise Scope Alarms

Alarm View Automatic Refresh : 0 seconds

2. Choose how you want the tree view of your alarm instances to look in the alarm-summary window by selecting values from the two drop-down list boxes.

   In the default tree view, alarm instances are organized first by server and then by severity as shown below.



Alarms View
All Alarms (67)
flotsam (67)
Critical (61)
Major (0)
Minor (0)
Warning (0)
Inform (0)
Special (0)
Saturated (0)
VeryHigh (0)
High (0)
Medium (0)
Low (0)
VeryLow (5)
Normal (0)

To choose some other organization, change the values of one or both list boxes. The choices in each list box are **Server**, **Severity**, **Property Group**, and **Partition**. (A partition is a filter set up using the NerveCenter Web Client. Each partition can include a list of IP address ranges. For information on how to create a partition using the NerveCenter Web Client, see *Connecting to a Server* on page 51.)

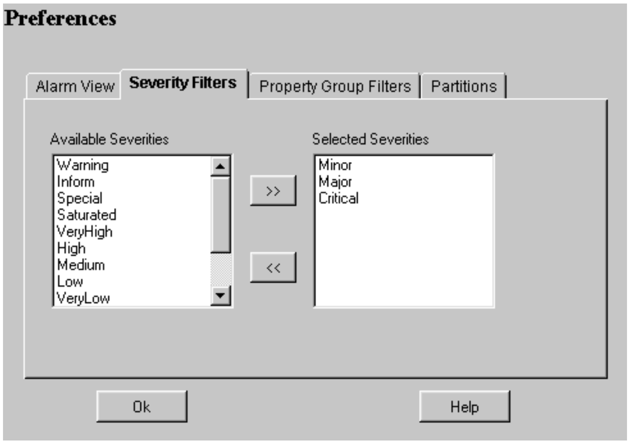
NOTE

Your specification of a tree view affects not only the organization of the tree view, but also how alarm instances are filtered. Instances are always filtered by server; however, they are filtered by severity only if you choose **Severity** from one of the listboxes. The same is true for filtering by property group or partition.

3. If you selected **Severity** in one of the list boxes, select the severities you want to use in filtering alarm instances. That is, only instances of the severities you select will appear on the alarm-summary page.

   a. Select the **Severity Filters** tab.

      The Severity Filters tab displays.



The first time you open it, the **Available Severities** list contains all the severities defined in the database of the first NerveCenter server to which you connected.

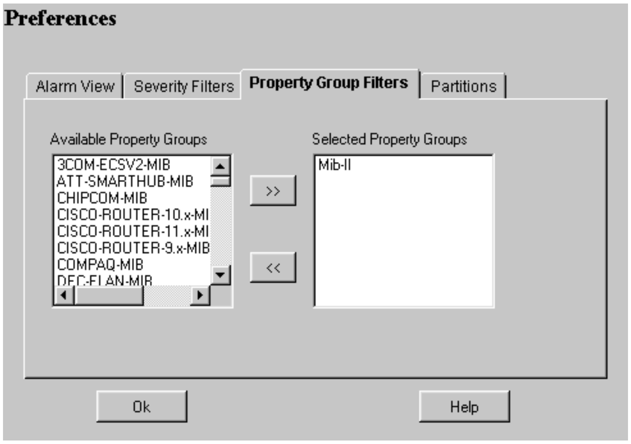
b. For each severity you want to use in your filtering, select the severity and then select the >> button.

The name of the severity displays in the **Selected Severities** list. Information about alarm instances with this severity will be displayed on the alarm-summary page.

4. If you selected **Property Group** in one of the list boxes, select the property groups you want to use in filtering alarm instances. That is, only alarm instances monitoring nodes in the property groups that you select will appear on the alarm-summary page.

a. Select the **Property Group Filters** tab.

The Property Group Filters tab is displayed.



The first time you open it, the **Available Property Groups** list contains the union of the property groups defined for each NerveCenter server to which you're connected.

b. For each property group you want to use in your filtering, select the property group and then select the >> button.

The property group displays in the **Selected Property Groups** list. Information about alarm instances monitoring a node in this property group will be displayed on the alarm-summary page.

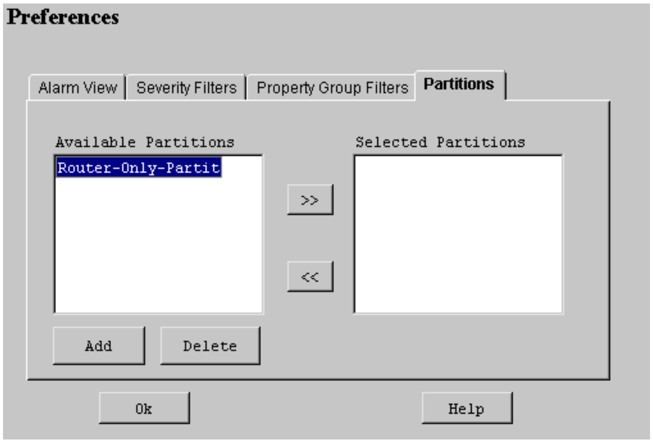
5. If you selected Partition in one of the list boxes, select the partitions you want to use in filtering alarm instances. That is, only instances monitoring machines on subnets specified in the partitions you select will appear on the alarm-summary page.

NOTE

If you have not defined any partitions, the **Available Partitions** list will be empty. Before you can perform this step you must define one or more partitions using the procedure describe in the section *Connecting to a Server* on page 51.

a. Select the **Partitions** tab.

The Partitions tab displays.



b. For each partition you want to use in your filtering, select the partition and then select the >> button.

   The partition name displays in the **Selected Partitions** list. Information about alarm instances monitoring nodes in this partition will be displayed on the alarm-summary page.

   See *Connecting to a Server* on page 51, for more information.

6. Check the **Display Enterprise Scope Alarms** checkbox if you want the Web client to display instances of enterprise scope alarms.

7. In the **Alarm View Automatic Refresh** field, enter a number of seconds, or leave the field set to 0.

   If you enter a nonzero value *x*, the Web client will refresh the alarm-summary page every *x* seconds. If you leave the value set to 0, the client will not refresh the page.

8. Select the **OK** button.

   Your preferences take effect, and you are taken to the alarm-summary page.

Defining a Partition

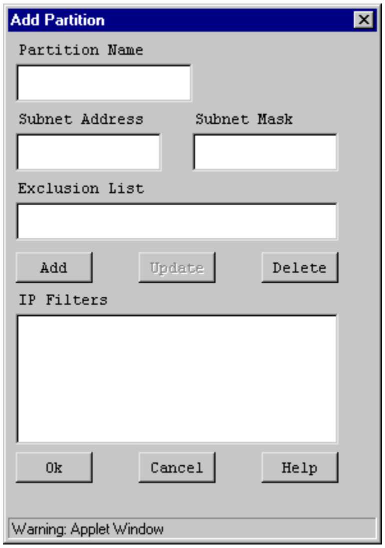
A Web-client partition is an alarm-instance filter that enables you to monitor the machines on one or more subnets. You also have the ability to monitor all of the machines on a subnet except ones that you explicitly exclude.

See *IP Subnet Filter Exclusion Rules* on page 66, for more information about filtering alarms by one or more subnets.

To create a partition

1. Go to the Preferences page.
2. Choose the **Partitions** tab.
3. Select the **Add Partition** button.

   The Add Partition dialog is displayed.



4. Type the name you want to give the partition in the **Partition Name** field.
5. In the **Subnet Address** and **Subnet Mask** fields, enter the subnet address and mask for the subnet you want to monitor. Both entries should contain four octets separated by periods.
6. In the **Exclusion List**, enter a comma-separated list of machines (or ranges of machines) on the subnet that you do not want to monitor.

For each machine, enter the last octet of its IP address. For example, the entry 1,2, 5-7 would mean to exclude the machines whose addresses end in 1, 2, 5, 6, and 7.

7. Select the **Add** button.

   The IP address filter displays in the **IP Filters** list.

8. Repeat step 5 to step 7 to add further subnets to the partition.

9. Select the **OK** button.

For the partition filter to take effect, you must:

- Add the partition to the **Selected Partitions** list on the **Partitions** tab
- Go to the **Alarm View** tab, and select **Partition** from one of the list boxes

Disconnecting from a Server

Unlike the NerveCenter Client, the NerveCenter Web Client does not offer a disconnect-from-server button or menu entry. To disconnect the Web client from a server, you must return to the Server Selection page, and remove the server from the **Selected Servers** list.

Disconnecting from a server from the Alarm Summary page

1. Select **Modify NerveCenter Server List** from the drop-down list box in the upper right corner of the window; then, select the **Go** button.

   You are taken to the Server Selection page.

2. Select a server from the **Selected Servers** list.

   This is the server you no longer want to connect to.

3. Select the << button.

4. Select the **OK** button.

   You are disconnected from the server you selected earlier and returned to the alarm-summary page.

Getting Started with NerveCenter Client

4

Before you can begin monitoring your network using the NerveCenter Client, you must start the client and then establish a connection between the client and one or more NerveCenter servers. You may also want to set up alarm filters to control which alarm instances the NerveCenter Client will display information about.

For instructions on how to perform these and related tasks, see the sections listed below:

Section	Description
Starting the Client on page 50	Describes how to start the NerveCenter Client.
Connecting to a Server on page 51	Explains how to log on to one or more NerveCenter Servers, discusses the various server connection options, and describes how to select an active server.
Setting Up Alarm-Instance Filters on page 61	Provides instructions for setting up alarm viewing preferences. You can request that the alarm instances from the servers you're connected to be filtered by: IP range, severity, or property group.
Specifying Heartbeat Messaging on page 81	Explains heartbeat messaging: how to set message intervals and how to deactivate heartbeat messaging.
Disconnecting from a Server on page 84	Describes how to log off the NerveCenter Server.

Starting the Client

The NerveCenter Client enables you to monitor current alarm instances, view an alarm’s history, reset an alarm, and monitor the status of nodes.

TO START THE CLIENT

- If you’re working on a UNIX system, from a terminal window, enter the command:

  ```
  client &
  ```

  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
  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 *Running the NerveCenter Server on UNIX* in *Managing NerveCenter*.

- If you’re working on a Windows system, start the client using the **Start** menu. If the person who installed NerveCenter selected the default program folder, NerveCenter, select the following set of menu entries: From the **Start** menu, select **Programs**, then **OpenService** NerveCenter, then **Client**.

  If the installer used a program folder other than Open NerveCenter, select **Client** from that folder instead.

  After you perform this step, you see the client window shown in Figure 4-1.

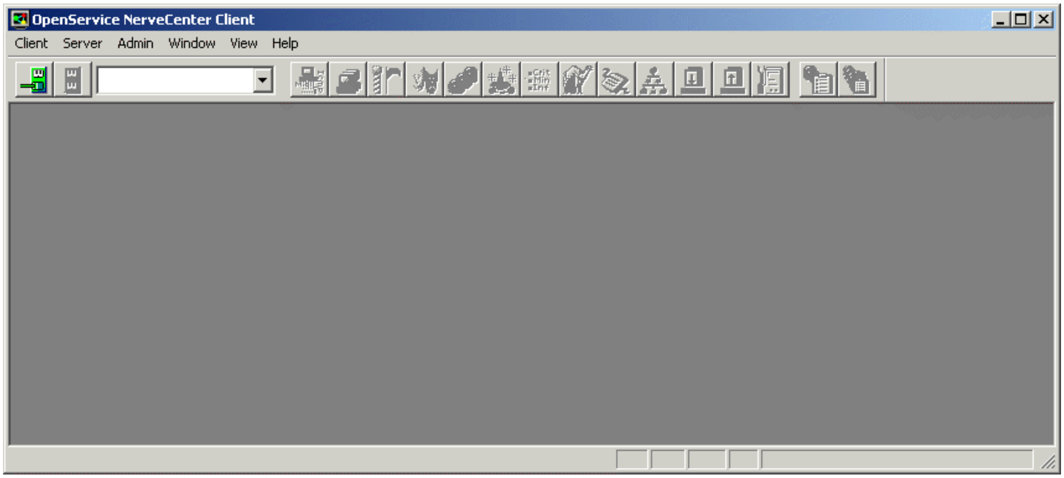


FIGURE 4-1. NerveCenter Client

Most of the buttons on the button bar and the options on the menus are not enabled until you connect the client to a NerveCenter server.

Connecting to a Server

Before you can use the client, you must connect the client to a NerveCenter server. This server collects data from managed devices, creates alarm instances, and performs the actions defined in alarms. The server also gives the client access to information about alarm instances and the status of nodes.

You can connect your client to more than one server at one time and view information about all the active alarm instances being managed by those servers. However, only one server can be the *active* server. The active server determines which NerveCenter database is used when you ask for a list of polls or the definition of an alarm.

For information on how to establish a connection with a NerveCenter server, see the following subsections:

- *Connecting to a Server Manually* on page 52
- *Connecting to a Server Automatically* on page 55
- *Sharing MIB Information from Multiple Servers* on page 57

You may also be interested in the following topics, which relate to connecting to a server:

- *Selecting the Active Server* on page 58
- *Deleting a Server from the Server List* on page 59
- *Changing the Client's Server Port* on page 60

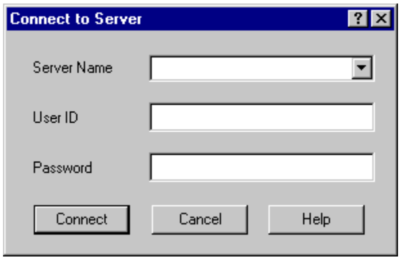
Connecting to a Server Manually

If you haven't configured the client to connect to one or more servers at startup, or if you want to establish a connection with a server that you don't typically use, you must establish your connection with the server manually.

To connect to a NerveCenter server manually

1. From the **Server** menu, select **Connect**.

   The Connect to Server window displays.



2. In the **Server Name** field, type the hostname or IP address of the machine where the NerveCenter server is running. Or choose a hostname or IP address from the **Server Name** drop-down list.

   The first time you connect to a server, the drop-down 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 won't display the names of machines to which you're already connected.) For information on removing an entry from the drop-down list box, see the section *Deleting a Server from the Server List* on page 59.

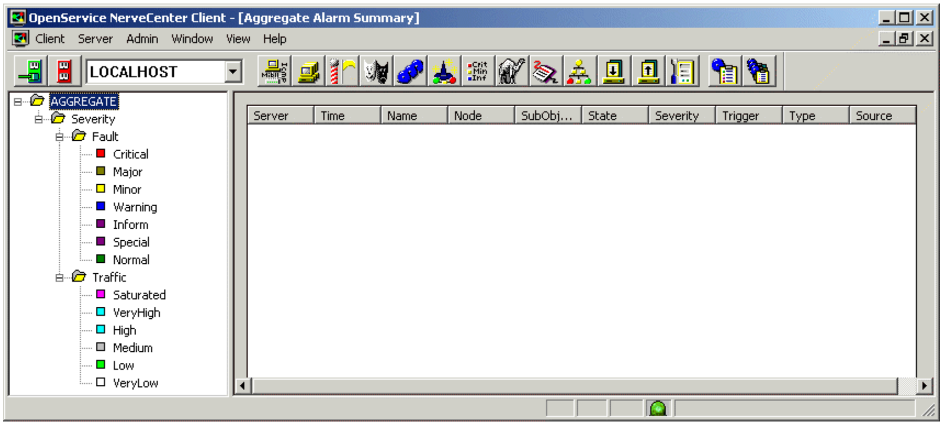
3. Type a user name and password in the **User ID** and **Password** fields.

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

4. Select the **Connect** button.

If the machine to which you try to connect is not running the NerveCenter server, you see the message **The server did not respond**.

When the client successfully connects to the server, all of the buttons in the button bar become enabled, and the Aggregate Alarm Summary window appears.



Server	Time	Name	Node	SubObj...	State	Severity	Trigger	Type	Source

FIGURE 4-2. Client Connected to a Server

Table 4-1 lists the client windows you can reach by using the buttons in the client's toolbar.

TABLE 4-1. Windows Accessible from Toolbar

Button	Window
🔌	Opens the Connect to Server window. From this window, you can connect the client to a NerveCenter server.
🔌	Opens a Client message window containing the prompt **Disconnecting from Hostname**. Use this window to confirm that you want to disconnect the client from a NerveCenter server.
MIB	Opens the Property Group List window. From this window, you can view the currently defined property groups and the properties that each property group contains.

TABLE 4-1. Windows Accessible from Toolbar (Continued)

Button	Window
	Opens the Node List window. From this window, you can view a list of the nodes defined in the NerveCenter database and a brief definition of each node.
	Opens the Poll List window. From this window, you can view a list of the polls defined in the NerveCenter database and a brief definition of each poll.
	Opens the Mask List window. From this window, you can view a list of the trap masks defined in the NerveCenter database and a brief definition of each trap mask.
	Opens the Alarm Definition List window. From this window, you can view a list of the alarms defined in the NerveCenter database and open a definition window for each alarm.
	Displays a list of currently defined correlation expressions. Correlation expressions enable you to create alarms from boolean expressions.
🟥Crit 🟨Min 🟦Inf	Opens the Severity List window. From this window, you can view a list of the severities defined in the NerveCenter database. (A severity has a name, a severity level, and a color associated with it.)
	Opens the Perl Subroutine List window. From this window, you can view a list of the currently defined Perl subroutines.
	Opens the Report List window. From this window, you can view a list of reports.
	Opens the Action Router Rule List window. From this window, you can view a list of the current set of rules that you have defined for the Action Router.
	Opens the Import Objects and Nodes dialog. From this dialog, you can import behavior models from one NerveCenter to another.
	Opens the Export Objects and Nodes dialog. From this dialog, you can export specific objects or groups of objects from one database to another.
	Opens the Server Status dialog. This dialog provides you with a comprehensive view of all your NerveCenter server settings.
	Opens the Alarm Summary window. This window presents information about the current alarm instances for the active server.
	Opens the Aggregate Summary window. This window presents information about the current alarm instances for all the servers to which you’re connected.

Connecting to a Server Automatically

If you want to establish a connection with the same set of servers each time you run the client, you can use NerveCenter's Autoconnect feature.

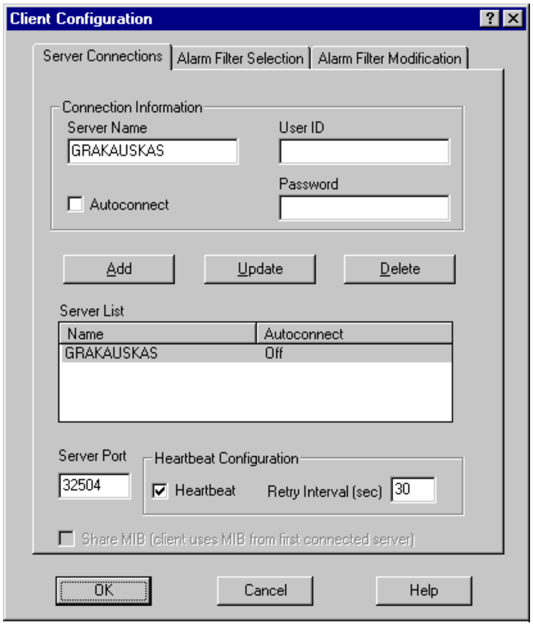
TIP

Before you activate the Autoconnect feature, you might want to manually connect to the NerveCenter Server, to verify that you can indeed access the server.

TO SET UP A LIST OF SERVERS TO WHICH YOU'LL CONNECT AT STARTUP

1. From the client's **Client** menu, choose **Configuration**.

   The Client Configuration dialog displays.



Server Name: GRAKAUSKAS

User ID:

Password:

□ Autoconnect

Name	Autoconnect
GRAKAUSKAS	Off

Server Port: 32504

☑ Heartbeat   Retry Interval (sec): 30

□ Share MIB (client uses MIB from first connected server)

2. Enter the hostname or IP address of the server to which you want to connect in the **Server Name** field.

3. Generally, you'll leave the default value in the **Server Port** field.

However, if the administrator who configured the server you want to connect to has changed the server port to be used for client/server communication, you must enter the new port number here. The NerveCenter Client uses this same port number for every NerveCenter Server to which it attempts to connect.

4. Check the **Autoconnect** checkbox.
5. Type a user name and password in the **User ID** and **Password** fields.

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

   On UNIX, if you have activated Autoconnect and your password changes, you must manually update your password in the Client Configuration dialog box for the Autoconnect feature to work. For the Autoconnect feature, NerveCenter does not update your password automatically.
6. Select the **Add** button.

   The server's name and automatic-connection status are displayed in the list near the bottom of the window.
7. Repeat step 2 through step 6 for each server you want to connect to automatically.
8. Select the **OK** button.

When you restart and log on to the client, you will be connected to the servers that have an Autoconnect status of On. Alternatively, you can connect, or reconnect, to these servers by selecting **Autoconnect** from the client's **Server** menu.

Sharing MIB Information from Multiple Servers

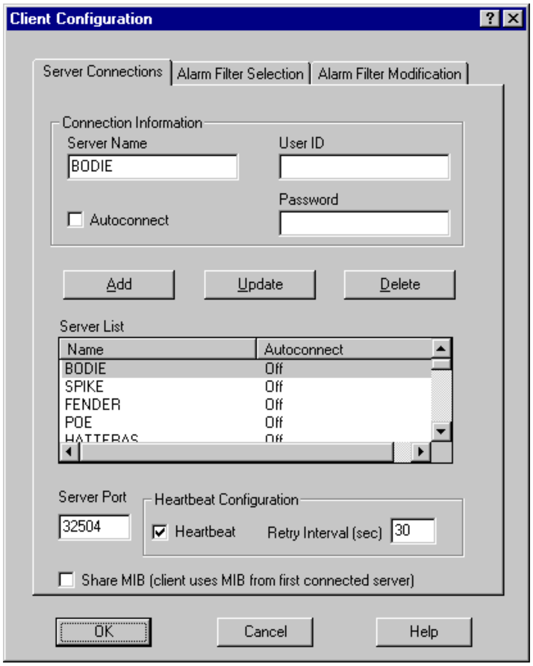
The NerveCenter Client needs a copy of the same MIB file that a NerveCenter Server uses to provide MIB base objects and attributes. If you intend to connect to multiple servers that use the same MIB file, you can direct NerveCenter to share MIB information. When you use this option, the NerveCenter Client saves only the MIB information sent to it by the first connected server.

For more information about MIBs, refer to *Managing Management Information Bases (MIBs)* in *Managing NerveCenter*.

To share MIB information

1. Disconnect from any connected servers.
2. From the client's **Client** menu, choose **Configuration**.

   The Client Configuration dialog is displayed.



Connection Information

Server Name: BODIE

User ID:

Password:

□ Autoconnect

Server List

Name	Autoconnect
BODIE	Off
SPIKE	Off
FENDER	Off
POE	Off
HATTERAS	Off

Server Port: 32504

Heartbeat Configuration

☑ Heartbeat Retry Interval (sec) 30

□ Share MIB (client uses MIB from first connected server)

3. Select the **Share MIB** checkbox.

4. Select the **OK** button.

Selecting the Active Server

The active server is the one whose database you can read data from. That is, you have access to this server's alarm definitions, poll definitions, and so on. You can view alarm instances for any number of servers at the same time.

TO MAKE A PARTICULAR SERVER THE ACTIVE SERVER

1. Display the server drop-down list on the client's button bar.

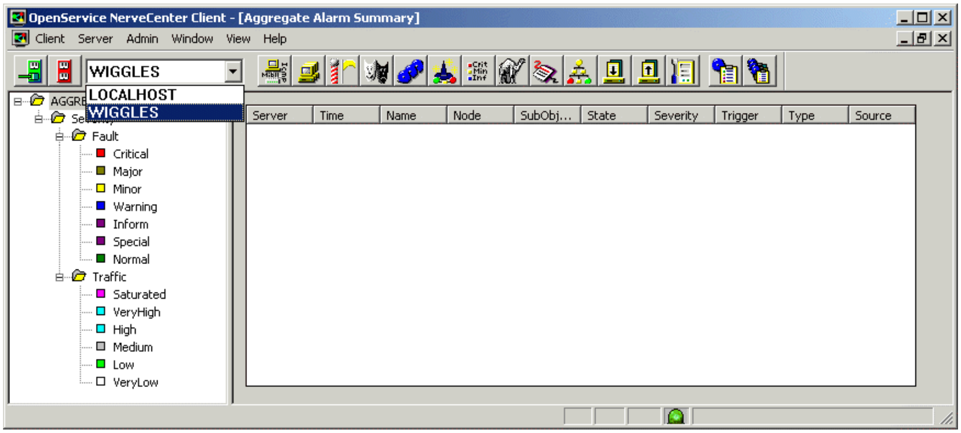


FIGURE 4-3. Server Drop-Down List

2. Select from the list the name of the server you want to make the active server.

   The name of the active server appears in the drop-down list box.

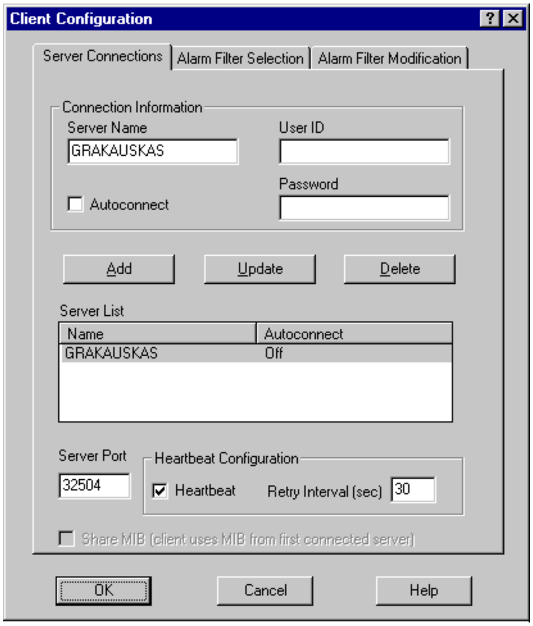
Deleting a Server from the Server List

NerveCenter maintains a list of servers that a client has connected to, or attempted to connect to, in the past. This list is used in the Connect to Server window, which you use to establish a connection to a server manually, and it also appears in the Client Configuration window. This list may contain the names of servers that you will never connect to again, or, even worse, the misspelled names of servers you were unable to connect to because of a misspelling.

To delete the name of a server from the server list

1. From the client's **Client** menu, select **Configuration**.

   NerveCenter's Client Configuration window is displayed.



Client Configuration

Server Connections | Alarm Filter Selection | Alarm Filter Modification

Connection Information

Server Name: GRAKAUSKAS

User ID:

Password:

□ Autoconnect

Server List

Name	Autoconnect
GRAKAUSKAS	Off

Server Port: 32504

Heartbeat Configuration

☑ Heartbeat

Retry Interval (sec): 30

□ Share MIB (client uses MIB from first connected server)

2. In the **Server List** near the bottom of the window, select the server name you want to remove from the server list.
3. Select the **Delete** button.
4. Select the **OK** button.

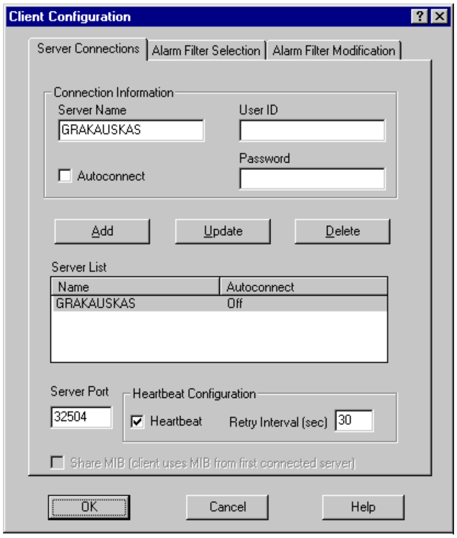
Changing the Client’s Server Port

Each NerveCenter server uses a special port on its host for client/server communication. By default, servers use port 32504; however, the person who configures the NerveCenter server can change the number of this communication port if port 32504 is being used by another application. If this number is changed on the server side, you must make a corresponding change on the client side before you will be able to connect to the server.

To change the client’s server port

1. From the client’s **Client** menu, choose **Configuration**.

   The Client Configuration window is displayed.



Client Configuration

Connection Information

Server Name: GRAKAUSKAS

User ID:

Password:

□ Autoconnect

Server List

Name	Autoconnect
GRAKAUSKAS	Off

Server Port: 32504

Heartbeat Configuration

☑ Heartbeat

Retry Interval (sec): 30

□ Share MIB (client uses MIB from first connected server)

2. In the **Server List** near the bottom of the window, select the name of the server that uses the non-default port number.

   Connection information for that server is displayed.

3. Type the new port number in the **Server Port** text field.
4. Select the **OK** button.

Setting Up Alarm-Instance Filters

Before or after you’ve connected to the servers from which you want to retrieve alarm instances, you can set up one or more alarm-instance filters, per server. These filters control which alarm instances are displayed in the NerveCenter Client. You can filter alarm instances by:

- The IP address of the instance’s node
- The severity of the instance’s state
- The property group associated with the instance’s node

If you filter alarm instances by a severity, only instances whose states have this severity will be displayed in the client. Filters based on property groups and IP address ranges work similarly.

A single filter can contain any combination of:

- A list of subnets
- A list of severities
- A list of property groups

These filters offer two advantages. First, they limit the number of alarm instances that will show up in the client, enabling you to focus your attention on the alarm instances that are specifically of interest to you. Using filters also improves the performance of the client, since NerveCenter only transfers to the client those alarm instances that match the filter criteria.

For information on how to build an alarm-instance filter and on how to associate a filter with a server, see the sections listed below:

- *Filtering Alarms by IP Range* on page 62
- *Filtering Alarms by Severity* on page 70
- *Filtering Alarms by Property Groups* on page 74
- *Associating a Filter with a Server* on page 78
- *Rules for Associating Filters with Alarms* on page 80

Filtering Alarms by IP Range

When you filter alarms by IP range, you are specifying that you only want to display alarm instances in the NerveCenter Client from particular nodes identified by their IP addresses.

See *IP Subnet Filter Exclusion Rules* on page 66, for more information about filtering alarms by IP ranges.

Although you can create a filter simply based on an IP range, a single filter can contain any combination of:

- A list of subnets
- A list of severities
- A list of property groups

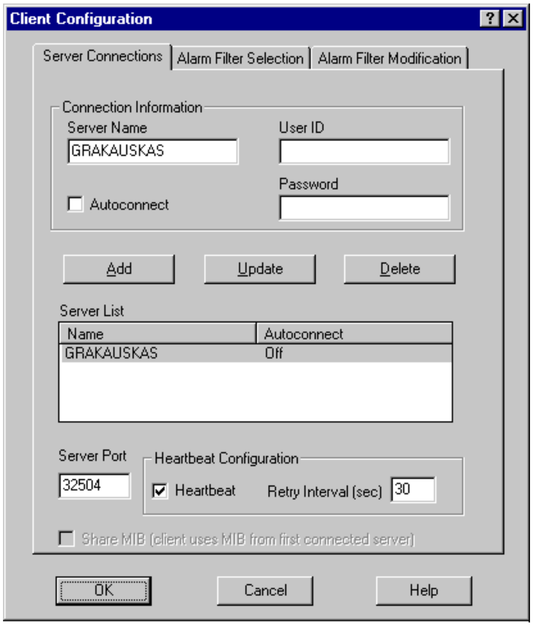
For information on how to build an alarm-instance filter based on severities and property groups, see the respective section listed below:

- *Filtering Alarms by Severity* on page 70
- *Filtering Alarms by Property Groups* on page 74

TO CREATE AN ALARM FILTER BASED ON AN IP RANGE

1. Choose **Configuration** from the **Client** menu.

   The Client Configuration dialog is displayed.



Server Name: GRAKAUSKAS

User ID:

Password:

□ Autoconnect

Server List

Name	Autoconnect
GRAKAUSKAS	Off

Server Port: 32504

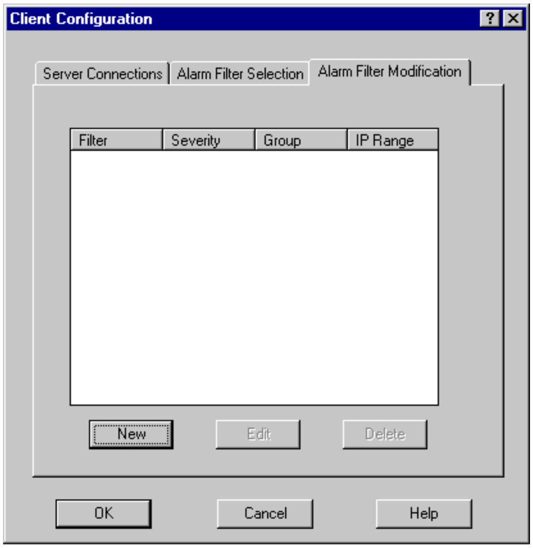
Heartbeat Configuration

☑ Heartbeat Retry Interval (sec) 30

□ Share MIB (client uses MIB from first connected server)

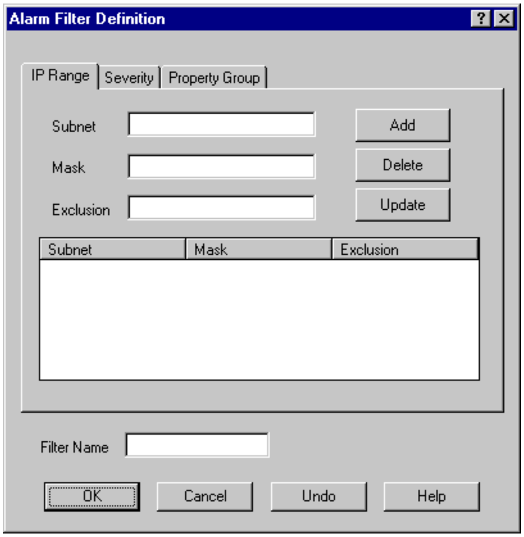
2. Select the **Alarm Filter Modification** tab.

   The Alarm Filter Modification page is displayed.



3. Select the **New** button.

   The Alarm Filter Definition dialog is displayed.



   This is the dialog you use to define your filter.

4. If you want to filter alarm instances based on the IP addresses of the alarm instances' nodes, perform the steps below for each subnet you want to be part of the filter. That is, you want to see information about instances whose nodes have IP addresses on these subnets.

    a. Enter an IP address in the **Subnet** text field.

       The IP address must consist of four octets separated by periods.

    b. Enter a subnet mask in the **Mask** text field.

       The subnet mask must consist of four octets separated by periods. Taken together with the subnet address, this mask defines the subnet whose nodes you're monitoring.

    c. In the **Exclusion** text field, enter the last octet of the IP address of any node on the subnet that you're not monitoring.

       You can enter multiple exclusions separated by commas. You can also enter a range of excluded nodes using a hyphen. For example, if you enter 24, 76-78 in the Exclusion field, the nodes whose addresses end in 24, 76, 77, and 78 will be excluded by the filter.

    d. Select the **Add** button.

    e. Repeat step a to step d to add other subnets to the alarm filter.

5. Enter a name for your filter in the **Filter Name** field.

6. Select the **OK** button.

   The Alarm Filter Definition dialog is closed and you return to the Client Configuration dialog box.

You've now defined an alarm filter based on an IP range. Before the client will use the filter, however, you must associate the filter with a server. For instructions on how to create this association, see the section *Associating a Filter with a Server* on page 78.

IP Subnet Filter Exclusion Rules

When you filter by subnet, you specify which subsets of nodes are managed by NerveCenter. Filtering does not apply to nodes that have been imported from a file or from another NerveCenter. For an example, see *IP Subnet Filter Examples* on page 68.

You can exclude specific nodes that belong to the filter by entering an exclusion. To exclude one or more nodes, you must specify the full subnet and mask, and then enter the individual nodes you want excluded. Enter the part of the IP address that is not affected by the subnet's mask.

NerveCenter filters Class B and C networks.

Class C Networks

In a Class C network, the first three octets of the address specify the network and the last octet specifies the host. For example, in network 194.123.45.0, the 194.123.45 value pertains to the network. The remaining octet is used to identify nodes (up to 254) on the network, and you can exclude nodes by specifying ID values in this octet.

Class B Networks

For a Class B network, only the first two octets of the address specify the network. For example, in network 132.45.0.0, the 132.45 value pertains to the network. The remaining two octets are used to identify nodes, and you can exclude nodes by specifying ID values in these two octets.

Example

In the following example, the node whose IP address is 134.204.179.40 is excluded from the filter (the node is filtered out and, therefore, is not managed by NerveCenter).

```
134.204.179.0
255.255.255.0
40
```

Rules for Exclusions

- You can enter several nodes separated by a comma. NerveCenter accepts comma-separated values with or without spaces following the commas. You can enter the node values in any order.

  The following three examples (each on a separate line) illustrate valid exclusions:

  ```
  7,8,9,15
  7, 8, 9, 15
  8,7,9,15
  ```

- You can enter a range of values using a hyphen.

  For example, you can enter as an exclusion range: **40-60**

  You can also enter the range in inverse order: **60-40**

- You can include multiple entries for the same subnet if you have values or ranges that are not incremental.

  - For example, you can enter as a filter:

    ```
    134.204.179.0
    255.255.255.0
    7,8,9
    134.204.179.0
    255.255.255.0
    40-60
    134.204.179.0
    255.255.255.0
    70-90
    ```

  - You can combine ranges, for example:

    ```
    134.204.179.0
    255.255.255.0
    40-60,70-90
    ```

  - You can also combine formats, for example:

    ```
    134.204.179.0
    255.255.255.0
    7-9,31,33,40-60
    ```

IP Subnet Filter Examples

The following examples can help you understand how to filter nodes for Class B and C networks.

Class C Network

The following subnet filters are for two sample nodes:

- Sample node #1 with IP address: 197.204.179.25
- Sample node #2 with two IP addresses:
  - 134.204.179.40
  - 197.204.179.7

The filter values in Table 4-2have the following effects on the sample nodes:

TABLE 4-2. Class C Network Examples

Subnet Mask Exclusion	Results of Filter
134.204.179.0 255.255.255.0	This filter does not contain any exclusions. Node #1 is not on this subnet and is not included in the filter or managed by NerveCenter. Node #2 is included in the filter because it’s on the subnet.
134.204.179.0 255.255.255.0 25,40	Node #1 is not on this subnet and is not included in the filter. Node #2 is listed as an exclusion and is not included in the filter.
197.204.179.0 255.255.255.0 7-20	Node #1 is included. Node #2 is not included because it’s listed in the exclusion range.
197.204.179.0 255.255.255.0 7-20 134.204.179.0 255.255.255.0 40	Node #1 is included in the first subnet. Node #2 is not included because it’s listed as an exclusion on both subnets.
197.204.179.0 255.255.255.0 25,40	Node #1 is not included because it’s listed as an exclusion. Node #2 is included.

Class B Filters

The following subnet filters are for two sample nodes:

- Sample node #1 with IP address: 132.45.160.10
- Sample node #2 with IP address: 132.45.174.10

The mask you use for this filter is 255.255.0.0.

TABLE 4-3. Class B Filter Examples (Set One)

Subnet Mask Exclusion	Results of Filter
132.45.0.0 255.255.0.0	Both nodes are included in the filter and managed by NerveCenter.
132.45.0.0 255.255.0.0 174.10	Node #1 is included in the filter. Node #2 is excluded from the filter. The filter includes all nodes except 132.45.174.10.
132.45.0.0 255.255.0.0 160.10-174.5	Node #1 is listed in the exclusion range and is excluded from the filter. Note #2 is included in the filter.
132.45.0.0 255.255.0.0 10	Both nodes are excluded from the filter and, therefore, neither node is managed by NerveCenter. The filter includes all nodes except 132.45.xxx.10, where xxx can be any value greater than 1 and less than 255.

If you use a subnet mask of 255.255.240.0, you would get different results.

- Sample node #1 with IP address: 132.45.160.10
- Sample node #2 with IP address: 132.45.174.10

You must first apply the filter before determining the node's ID. The filter values in the table below have the following effects:

TABLE 4-4. Class B Filter Examples (Set Two)

Subnet Mask Exclusion	Results of Filter
132.45.160.0 255.255.240.0 174.10	The node is not included in the filter. The filter includes all nodes except 132.45.174.10.
132.45.160.0 255.255.240.0 10	Neither node is included in the filter. The filter includes all nodes except those ending in .10. The third octet of an excluded node can be 174 or any value between 160 and 174.

Filtering Alarms by Severity

When you filter alarms by severity, you are specifying that you only want to display alarm instances in the NerveCenter Client from particular nodes identified by the severity of the alarm instance's state.

Although you can create a filter simply based on severity, a single filter can contain any combination of:

- A list of subnets
- A list of severities
- A list of property groups

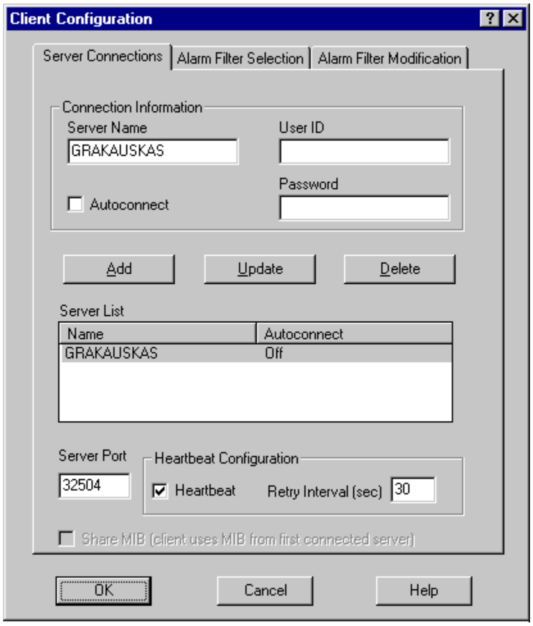
For information on how to build an alarm-instance filter based on IP range and property groups, see the respective section listed below:

- *Filtering Alarms by IP Range* on page 62
- *Filtering Alarms by Property Groups* on page 74

TO CREATE AN ALARM FILTER BASED ON SEVERITY

1. Choose **Configuration** from the **Client** menu.

   The Client Configuration dialog is displayed.



Client Configuration

Connection Information

Server Name: GRAKAUSKAS

User ID:

Password:

□ Autoconnect

Server List

Name	Autoconnect
GRAKAUSKAS	Off

Server Port: 32504

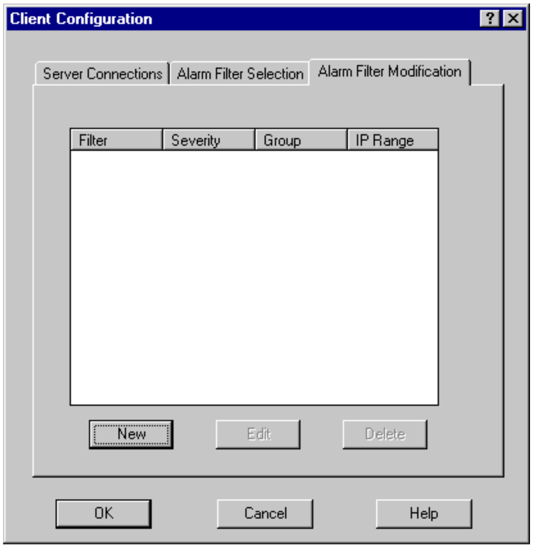
Heartbeat Configuration

☑ Heartbeat Retry Interval (sec) 30

□ Share MIB (client uses MIB from first connected server)

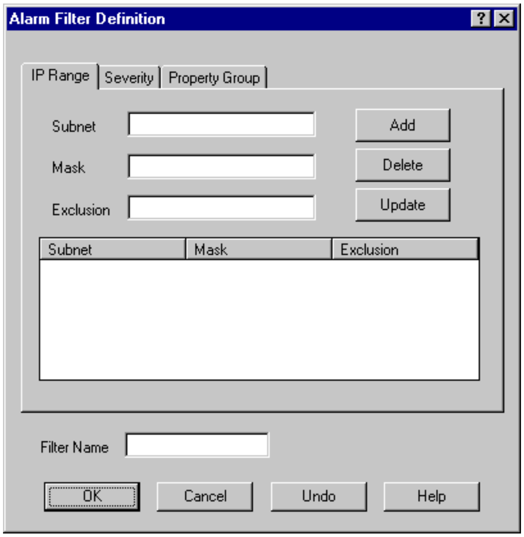
2. Select the **Alarm Filter Modification** tab.

   The Alarm Filter Modification page is displayed.



3. Select the **New** button.

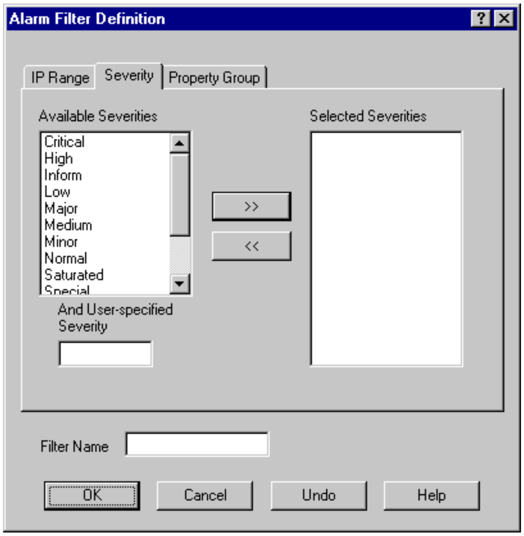
The Alarm Filter Definition dialog is displayed.



This is the dialog you use to define your filter.

4. Select the **Severity** tab.

The Severity tab is displayed.



5. In the **Available Severities** list, for each severity you want to use in your filter, select the severity and then select the >> button. That is, you want to see information about alarm instances whose states have these severities.

   The severities in this list box are the union of the severities defined by all of the servers to which you’re connected. Optionally, you can also add a user-defined severity to the list of severities to filter by entering a severity in the **And User-specified Severity** text box, and then selecting the >> button.

   The name of the severity is moved to the **Selected Severities** list. Information about alarm instances with this severity will be displayed in the alarm summary views.

   To remove a severity from the **Selected Severities** list, select the severity and then select the << button.

6. Enter a name for your filter in the **Filter Name** field.

7. Select the **OK** button.

   The Alarm Filter Definition dialog is closed and you return to the Client Configuration dialog box.

You’ve now defined an alarm filter based on severity. Before the client will use the filter, however, you must associate the filter with a server. For instructions on how to create this association, see the section *Associating a Filter with a Server* on page 78.

Filtering Alarms by Property Groups

When you filter alarms by property groups, you are specifying that you only want to display alarm instances in the NerveCenter Client from particular nodes belonging to one or more property groups.

Although you can create a filter simply based on membership within a property group, a single filter can contain any combination of:

- A list of subnets
- A list of severities
- A list of property groups

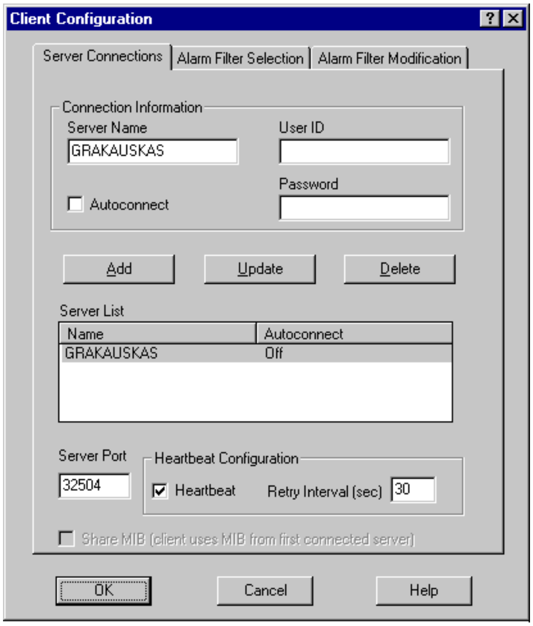
For information on how to build an alarm-instance filter based on an IP range and severities, see the respective section listed below:

- *Filtering Alarms by IP Range* on page 62
- *Filtering Alarms by Severity* on page 70

TO CREATE AN ALARM FILTER BASED ON PROPERTY GROUPS

1. Choose **Configuration** from the **Client** menu.

   The Client Configuration dialog is displayed.



Server Name: GRAKAUSKAS

User ID:

Password:

☐ Autoconnect

Name	Autoconnect
GRAKAUSKAS	Off

Server Port: 32504

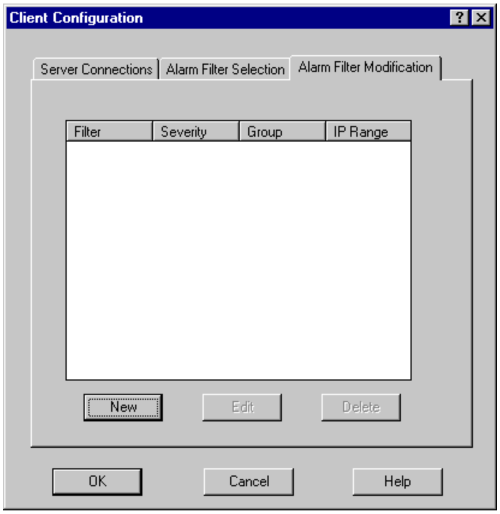
☑ Heartbeat

Retry Interval (sec): 30

☐ Share MIB (client uses MIB from first connected server)

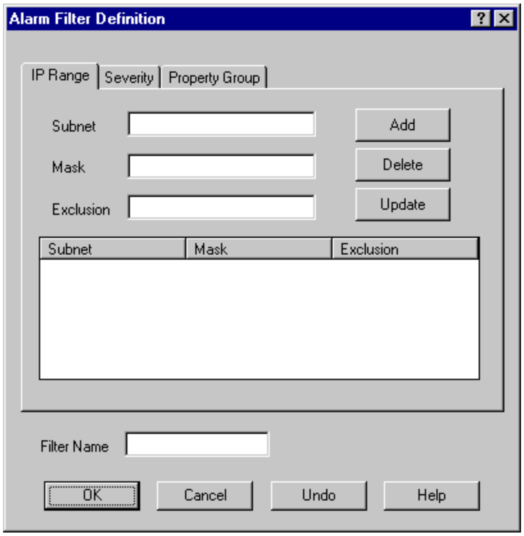
2. Select the **Alarm Filter Modification** tab.

   The Alarm Filter Modification tab is displayed.



3. Select the **New** button.

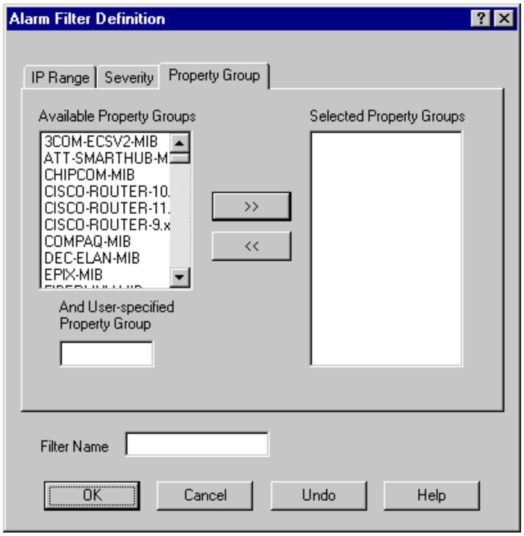
   The Alarm Filter Definition dialog is displayed.



   This is the dialog you use to define your filter.

4. Select the **Property Group** tab.

   The Property Group tab is displayed.



5. In the **Available Property Groups** list, for each property group of each alarm instance's node, perform the steps below for each property group you want to be part of the filter. That is, you want to see information about instances whose nodes belong to these property groups.

   The property groups in this list box are the union of the property groups defined by all of the servers to which you're connected.

   The name of the property group is moved to the **Selected Property Groups** list. Information about alarm instances with this property will be displayed in the alarm summary views. Optionally, you can also add a user-defined property group to the list of properties to filter by entering a property group in the **And User-specified Property Group** text box, and then selecting the >> button.

   To remove a property group from the **Selected Properties** list, select the property group and then select the << button.

6. Enter a name for your filter in the **Filter Name** field.

7. Select the **OK** button.

   The Alarm Filter Definition dialog is closed and you return to the Client Configuration dialog box.

You’ve now defined an alarm filter based on property groups. Before the client will use the filter, however, you must associate the filter with a server. For instructions on how to create this association, see the section *Associating a Filter with a Server* on page 78.

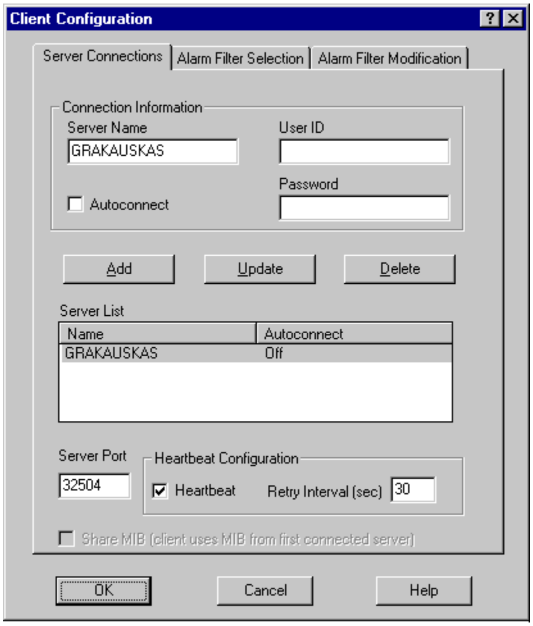
Associating a Filter with a Server

When you define an alarm filter, that filter is not used to filter alarm instances from all connected servers. It is only used to filter alarm instances from a server with which you have explicitly associated it.

To associate an alarm filter with a NerveCenter server

1. Choose **Configuration** from the **Client** menu.

   The Client Configuration dialog is displayed.



Connection Information

Server Name: GRAKAUSKAS

User ID:

Password:

□ Autoconnect

Server List

Name	Autoconnect
GRAKAUSKAS	Off

Server Port: 32504

Heartbeat Configuration

☑ Heartbeat Retry Interval (sec): 30

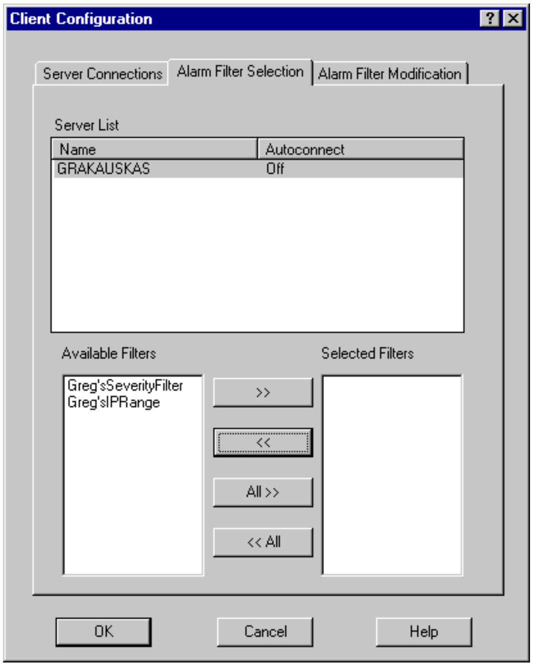
□ Share MIB (client uses MIB from first connected server)

2. Select a server from the list of servers at the bottom of the dialog.

   The name of the server appears in the **Server Name** text field in the Connection Information group box. This is the server with which you will associate your alarm filter.

3. Select the **Alarm Filter Selection** tab.

   The Alarm Filter Selection page is displayed.



Name	Autoconnect
GRAKAUSKAS	Off

Greg'sSeverityFilter
Greg'sIPRange

4. Select a filter from the **Available Filters** list.

   This is the filter you want to associate with the server you selected in step 2.

5. Select the >> button to move the filter from the **Available Filters** list to the **Selected Filters** list.

   To remove a filter from the **Selected Filters** list, select the filter and then select the << button.

6. Select the **OK** button at the bottom of the dialog.

Rules for Associating Filters with Alarms

When deciding whether to apply multiple filters to your alarms, you should keep in mind the following general rules:

- Multiple filters are ORed together
- Multiple conditions in a single filter are ANDed together

Multiple Filters are ORed Together

When you select more than one filter for a server, each filter is independent of the other filters. Their behavior is equivalent to a logical OR operation.

For example, say you associate two filters with a NerveCenter Server. The two filters are defined as follows:

- Filter #1 is configured to display only those alarms that have a severity level of Critical.
- Filter #2 is configured to display only those alarms coming from the network 132.168.196.0.

When both filters are applied to a server, you see the following alarms:

- Alarms with a Critical severity level from all existing networks defined for the server.
- From the network 132.168.196.0, you see all alarms regardless of severity.

Multiple Conditions in a Single Filter are ANDed Together

If, instead of the above view, you want to limit your alarms to Critical instances coming from the network 132.168.196.0, you need to create one filter with both of those conditions. You would create one filter that:

- Specifies a severity level of Critical, and
- Specifies an IP range of 132.168.196.0.

With this filter applied to the server, you see only those alarms that have a Critical severity level *and* that come from network 132.168.196.0. One filter with multiple conditions is equivalent to a logical AND operation; each condition is ANDed with the other conditions for optimum filtering.

Specifying Heartbeat Messaging

The NerveCenter Client sends a message called a *heartbeat* to each connected NerveCenter Server on a standard interval. This messaging ensures the reliability of communications between the server and client. If a server fails to respond after three consecutive heartbeat messages from the client, a message box is displayed on the client console to alert the operator of the server's heartbeat failure. (In such cases, you should check with your network administrator to obtain the status of that particular NerveCenter Server.)

You can set the interval at which the NerveCenter Client sends a heartbeat to the NerveCenter Server. (By default, this interval is 30 seconds.) You can also choose to deactivate heartbeat messaging.

See the following sections for more information:

- *Modifying the Heartbeat Message Interval* on page 81
- *Deactivating Heartbeat Messaging* on page 83

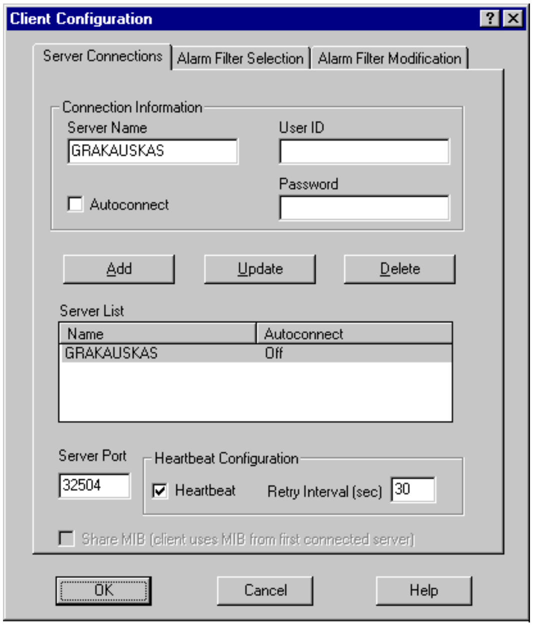
Modifying the Heartbeat Message Interval

You can change the interval NerveCenter Client uses to send heartbeat messages to verify its connection with your NerveCenter Servers.

TO MODIFY THE HEARTBEAT MESSAGE INTERVAL

1. Choose **Configuration** from the **Client** menu.

   The Client Configuration dialog is displayed.



Server Name: GRAKAUSKAS

User ID:

Password:

□ Autoconnect

Server List

Name	Autoconnect
GRAKAUSKAS	Off

Server Port: 32504

☑ Heartbeat Retry Interval (sec): 30

□ Share MIB (client uses MIB from first connected server)

2. In the **Heartbeat Configuration** panel, make sure the **Heartbeat** checkbox is checked. If it's not checked, heartbeat messaging is turned off.

3. In the **Retry Interval** field, enter the number of seconds you want NerveCenter Client to wait between heartbeat messages. The default is 30 seconds. (The number of retries is three.)

NOTE

When you modify heartbeat messaging, it applies to all NerveCenter Servers to which this client connects.

4. Select the **OK** button.

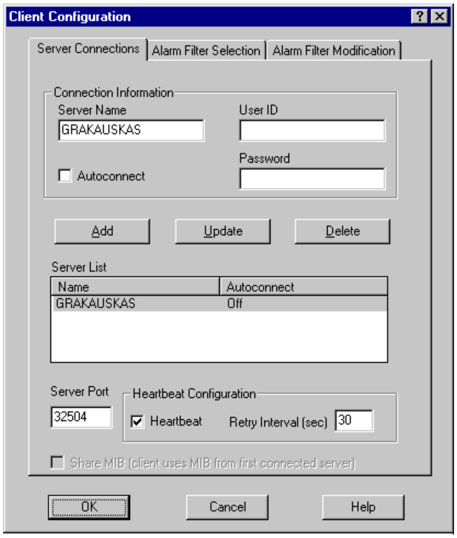
Deactivating Heartbeat Messaging

The NerveCenter Client sends heartbeat messages on an interval that you specify (or by default, every 30 seconds) to verify its connection with your NerveCenter Servers. If you choose, you can deactivate (or activate) heartbeat messages going to and from *all* your connected servers.

TO DEACTIVATE HEARTBEAT MESSAGES

1. Choose **Configuration** from the **Client** menu.

   The Client Configuration dialog is displayed.



Server Name: GRAKAUSKAS

User ID:

Password:

□ Autoconnect

Name	Autoconnect
GRAKAUSKAS	Off

Server Port: 32504

☑ Heartbeat Retry Interval (sec): 30

□ Share MIB (client uses MIB from first connected server)

2. In the Heartbeat Configuration panel, uncheck the **Heartbeat** checkbox.

NOTE

If there is no check mark in this checkbox, heartbeat messaging has already been deactivated for NerveCenter Client. When you activate or deactivate heartbeat messaging, it applies to all NerveCenter Servers to which this client connects.

3. Select the **OK** button.

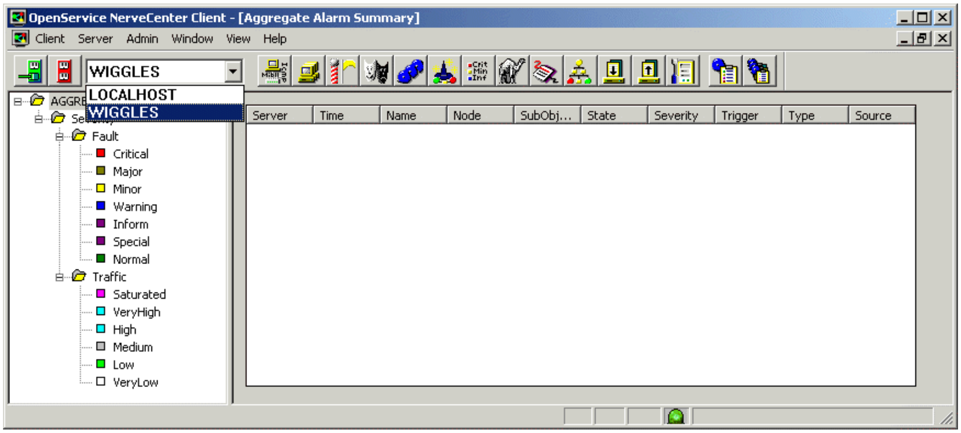
   Heartbeat deactivation takes effect the next time you connect NerveCenter Client to one or more of your NerveCenter Servers.

Disconnecting from a Server

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

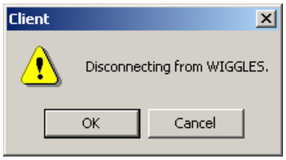
TO DISCONNECT THE CLIENT FROM A SERVER

1. From the server drop-down list on the client's button bar, select the server with which you want to break the connection.



2. From the client's **Server** menu, choose **Disconnect**.

   You see a pop-up window that asks you to confirm that you want to disconnect from the selected server.



Disconnecting from WIGGLES.

3. Select the **OK** button.

Monitoring Alarms

5

As you know, when NerveCenter detects a condition it is looking for, it creates an alarm instance that tracks that condition. For instance, if your site uses the behavior model that includes the alarm ifLoad, NerveCenter monitors network traffic on a set of interfaces. If it detects a traffic level above a certain threshold, it creates an alarm instance to track that condition.

Both the NerveCenter Web Client and the NerveCenter Client feature interfaces that enable you to monitor these alarm instances. This chapter discusses:

- The interfaces that these clients provide for monitoring alarm instances
- How to interpret the information these interfaces present
- How to examine an alarm instance's history

This information is presented in the following sections:

Section	Description
Viewing Alarm Information on page 86	Explains how to use the NerveCenter Web Client and the NerveCenter Client to track current alarm instances.
Interpreting Alarm-Instance Information on page 97	Explains how to use the NerveCenter Client to obtain additional information about the cause of an alarm transition.
Viewing an Alarm Instance's History on page 108	Explains how to use the NerveCenter Web Client or the NerveCenter Client to view an alarm instance's history.
Reading Logged Data on page 114	Explains how to read a log entry. This information is important if the alarm transition you're interested in has associated with it a Log to File or EventLog action.

Viewing Alarm Information

When an alarm instance is instantiated or undergoes a transition, you need to know certain things about the alarm transition that just took place:

- Which alarm was instantiated or underwent a transition? If the name of the alarm was ifLinkUpDown, you know that you received a link-down or a link-up trap.
- What node was the alarm instance monitoring? Was it monitoring a particular interface on a device?
- What state is the alarm instance now in? And what is the severity of that state? If the alarm involved is ifLinkUpDown and the current state is LinkDown (Major severity), you know that a communication link is down.
- What NerveCenter object caused the alarm instance to be instantiated or undergo a transition? If you know that the poll MediumLoad caused the instantiation or transition, you can look at the definition of that poll to determine exactly what condition is being reported.

Both the NerveCenter Web Client and the NerveCenter Client provide interfaces that present you with summary information about the current alarm instances in which you’re interested. For information about bringing up these interfaces and about the information they present, see the following sections:

- *Using the NerveCenter Web Client* on page 87
- *Using the NerveCenter Client* on page 92

Using the NerveCenter Web Client

To view information about current alarm instances using the NerveCenter Web Client, you use the Web client's alarm-summary window. By default, you are taken to this window when you log in.

TO GET TO THE ALARM-SUMMARY WINDOW VIA THE LOGIN PAGE

1. Enter your user name and password in the **Username** and **Password** fields.

NOTE

The username and password you enter have to be valid on the system that is running the Web Collector. Once validated, you can connect to any NerveCenter Server that has the same username/password combination configured as a valid NerveCenter user.

2. Make sure that the **Auto-Connect Previously Selected Servers** radio button is selected.

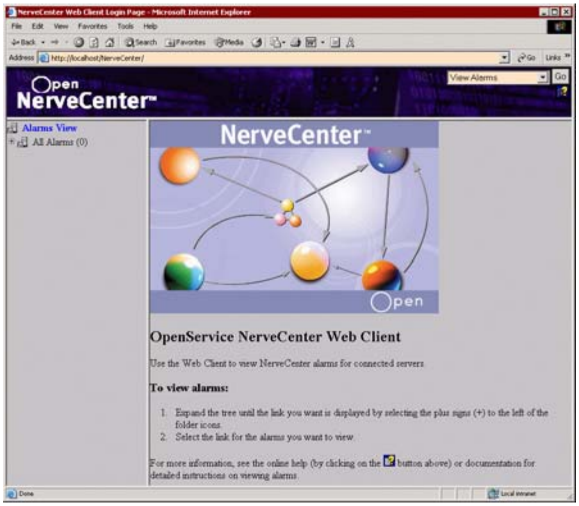
   Before you can connect to one or more NerveCenter servers and view information about the alarm instances created by those servers, you must have created a list of servers as described in *Modifying the Server Connection List* on page 40 and set your alarm viewing preferences as described in *Setting Preferences* on page 42.

3. Select the **OK** button.

NOTE

You are also taken to the alarm-summary window after you first select your servers and set your preferences.

The alarm-summary window displays, as shown in Figure 5-1.


NerveCenter Web Client Login Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/NerveCenter/
Go Links
Open NerveCenter
View Alarms
Go
Alarms View
All Alarms (0)
NerveCenter
Open
Done
Local intranet

OpenService NerveCenter Web Client

Use the Web Client to view NerveCenter alarms for connected servers

To view alarms:

1. Expand the tree until the link you want is displayed by selecting the plus signs (+) to the left of the folder icons.
2. Select the link for the alarms you want to view.

For more information, see the online help (by clicking on the button above) or documentation for detailed instructions on viewing alarms.

FIGURE 5-1. WebClient Alarm Summary View

The left frame in the window contains a tree view of the current alarm instances that meet the criteria you specified when you set your preferences. You can expand the tree by clicking on the plus sign associated with a branch that contains other branches or leaves, and you can display information about a set of alarm instances in the right frame by selecting one of the hypertext links in the tree view.

NOTE

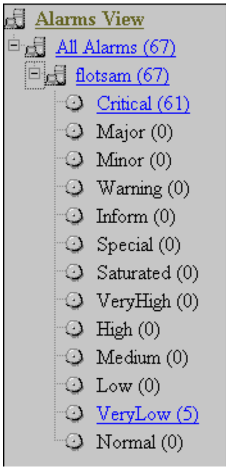
Since the left frame of the Web client does not does not update the tree view in real time, the number of alarms indicated in the alarm tree view might not always match the number of alarms shown in the right window frame. To view the most current count in the Alarm tree, select the browser's refresh button. You can also set a low refresh rate in the Web client's preferences. See the section *Setting Preferences* on page 42, for more information.

For information about how to interpret the information in tree and alarm-detail frames, see the following sections:

- *The Tree View* on page 89
- *The Alarm-Detail View* on page 90

The Tree View

The left frame in the alarm-summary window contains a tree view of the alarm instances you've requested to see. Figure 5-2 shows an expanded tree of alarm instances organized first by server, then by severity.



Alarms View
All Alarms (67)
flotsam (67)
Critical (61)
Major (0)
Minor (0)
Warning (0)
Inform (0)
Special (0)
Saturated (0)
VeryHigh (0)
High (0)
Medium (0)
Low (0)
VeryLow (5)
Normal (0)

FIGURE 5-2. Web Client Tree View

This tree view serves several purposes:

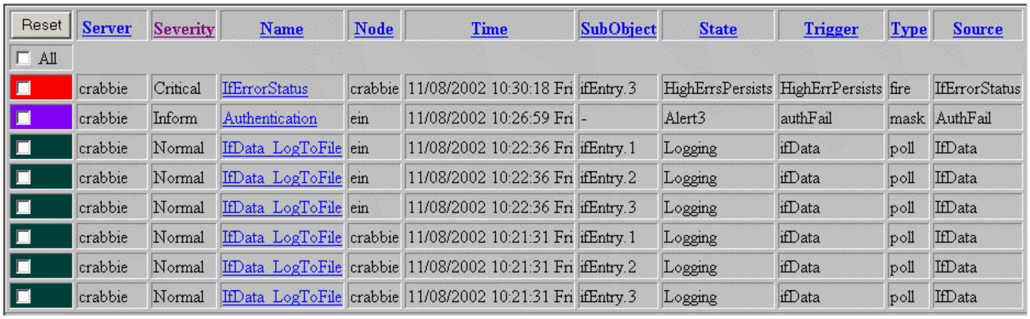
- The tree enables you to see at a glance the total number of current alarm instances and the number of instances in each instance group. For example, the figure above indicates that the Web client has retrieved 145 alarm instances from the server DURNCWEB and that 20 of these instances are of Minor severity.
- It enables you to view your alarm instances in different ways. By default, the instances are organized—as mentioned earlier—by server, then severity. However, you can organize the instances using any two of the following criteria:
  - Server
  - Severity
  - Property group
  - Partition

  You can change the organization of the tree by setting your preferences on the Preferences page. For instruction on how to do this, see the section *Setting Preferences* on page 42.

- The tree enables you to control which alarm instances appear in the alarm-detail frame. If you select the Major link, only alarm instances of Major severity will appear in the alarm-detail frame.

The Alarm-Detail View

The right frame in the alarm-summary window contains an alarm-detail view that presents quite a bit of information about selected alarm instances, as shown in Figure 5-3.



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

FIGURE 5-3. Alarm-Detail Frame

This is the frame you'll use for most of your monitoring.

Table 5-1 explains what information is available for each alarm instance.

TABLE 5-1. Fields in Alarm-Detail Pane

Column	Description
Name	The name of the alarm from which the alarm instance was created. If you have any question about what condition a particular alarm is monitoring, you can use the NerveCenter Client to view a definition of the alarm. For information about viewing such a definition, see the section *Getting Information about an Alarm* on page 98.
Node	The hostname or IP address of the node the alarm instance is monitoring.
Time	The date and time at which the alarm instance's most recent transition occurred.
SubObject	The subobject associated with the alarm instance. This subobject consists of a MIB base object plus an instance number, for example, ifEntry.1. The instance usually tells you which interface on a device is being monitored.
State	The current state of the alarm instance. The name of the state should indicate the condition NerveCenter is reporting. For example, if an instance of the alarm IfUpDownStatus is monitoring an interface and the current state of the alarm instance is "down," the operational status of the interface is down.

TABLE 5-1. Fields in Alarm-Detail Pane (Continued)

Column	Description
Trigger	The name of the trigger that caused the most recent alarm transition.
Type	The type of trigger that caused the most recent alarm transition. The possible types are poll, mask, fire (alarm), and built-in.
Source	The name of the poll, mask, or alarm that generated the trigger. Or, in the case of a built-in trigger, the name of the trigger. Given the name of the trigger that caused the transition and the name of the object that generated the trigger, you can pinpoint the exact cause of a transition. See the section *Getting Information about a Trigger* on page 100 for details on this subject.

The alarm-detail frame is designed primarily for reading. However, there are a couple of actions you can take from this frame.

- You can select any of the column headings to sort the alarm-instance entries alphabetically by the values in that column.

  This feature is useful for tasks such as ordering alarm instances by node.

- You can select an alarm name (a hypertext link) to open an alarm-history window. For more information about alarm history, see the section *Viewing an Alarm Instance's History* on page 108.

Using the NerveCenter Client

The NerveCenter Client provides two windows that you can use to view information about current alarm instances:

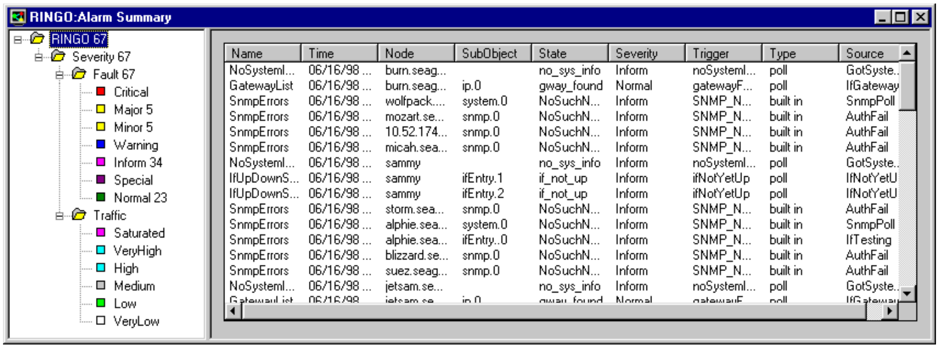
- The Alarm-Summary window
- The Aggregate Alarm Summary window

If you’re only connected to one server or are only interested in viewing alarm instances from one server at a time, you should use the Alarm Summary window. On the other hand, if you are connected to multiple servers and want to be able to view alarm instances from all servers at once, you should use the Aggregate Alarm Summary window.

To open the Alarm Summary window

- From the client’s **Admin** menu, choose **Alarm Summary**.

  The Alarm Summary window is displayed.



RINGO:Alarm Summary

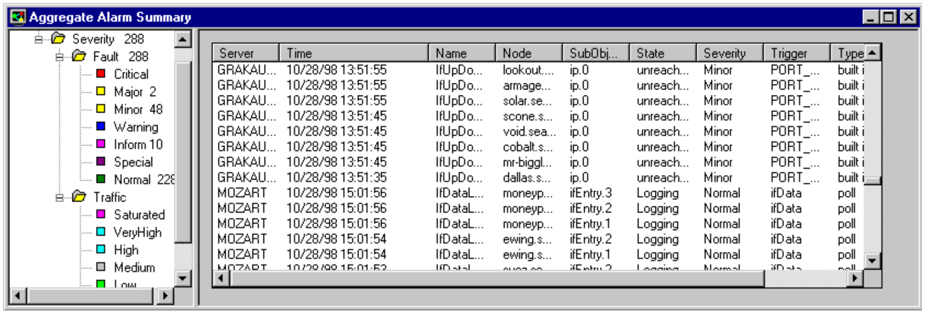
- RINGO 67
  - Severity 67
    - Fault 67
      - 🟥 Critical
      - 🟨 Major 5
      - 🟨 Minor 5
      - 🟦 Warning
      - 🟪 Inform 34
      - 🟪 Special
      - 🟩 Normal 23
    - Traffic
      - 🟪 Saturated
      - 🟦 VeryHigh
      - 🟦 High
      - ⬜ Medium
      - 🟩 Low
      - □ VeryLow

Name	Time	Node	SubObject	State	Severity	Trigger	Type	Source
NoSystemI...	06/16/98 ...	burn.seag...		no_sys_info	Inform	noSystemI...	poll	GotSyste...
GatewayList	06/16/98 ...	burn.seag...	ip.0	gway_found	Normal	gatewayF...	poll	IfGateway
SnmpErrors	06/16/98 ...	wolfpack....	system.0	NoSuchN...	Inform	SNMP_N...	built in	SnmpPoll
SnmpErrors	06/16/98 ...	mozart.se...	snmp.0	NoSuchN...	Inform	SNMP_N...	built in	AuthFail
SnmpErrors	06/16/98 ...	10.52.174...	snmp.0	NoSuchN...	Inform	SNMP_N...	built in	AuthFail
SnmpErrors	06/16/98 ...	micah.sea...	snmp.0	NoSuchN...	Inform	SNMP_N...	built in	AuthFail
NoSystemI...	06/16/98 ...	sammy		no_sys_info	Inform	noSystemI...	poll	GotSyste...
IfUpDownS...	06/16/98 ...	sammy	ifEntry.1	if_not_up	Inform	ifNotYetUp	poll	IfNotYetU
IfUpDownS...	06/16/98 ...	sammy	ifEntry.2	if_not_up	Inform	ifNotYetUp	poll	IfNotYetU
SnmpErrors	06/16/98 ...	storm.sea...	snmp.0	NoSuchN...	Inform	SNMP_N...	built in	AuthFail
SnmpErrors	06/16/98 ...	alphie.sea...	system.0	NoSuchN...	Inform	SNMP_N...	built in	SnmpPoll
SnmpErrors	06/16/98 ...	alphie.sea...	ifEntry..0	NoSuchN...	Inform	SNMP_N...	built in	IfTesting
SnmpErrors	06/16/98 ...	blizzard.se...	snmp.0	NoSuchN...	Inform	SNMP_N...	built in	AuthFail
SnmpErrors	06/16/98 ...	suez.seag...	snmp.0	NoSuchN...	Inform	SNMP_N...	built in	AuthFail
NoSystemI...	06/16/98 ...	jetsam.se...		no_sys_info	Inform	noSystemI...	poll	GotSyste...
GatewayList	06/16/98	jetsam.se	ip.0	gway_found	Normal	gatewayF	poll	IfGateway

TO OPEN THE AGGREGATE ALARM SUMMARY WINDOW

- From the client’s **Admin** menu, choose **Aggregate Alarm Summary**.

  The Aggregate Alarm Summary window is displayed.



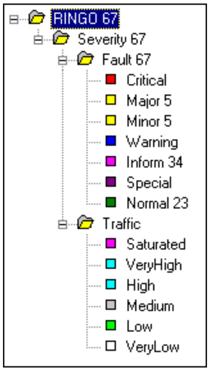
Server	Time	Name	Node	SubObj...	State	Severity	Trigger	Type
GRAKAU...	10/28/98 13:51:55	IfUpDo...	lookout....	ip.0	unreach...	Minor	PORT_...	built i
GRAKAU...	10/28/98 13:51:55	IfUpDo...	armage...	ip.0	unreach...	Minor	PORT_...	built i
GRAKAU...	10/28/98 13:51:55	IfUpDo...	solar.se...	ip.0	unreach...	Minor	PORT_...	built i
GRAKAU...	10/28/98 13:51:45	IfUpDo...	scone.s...	ip.0	unreach...	Minor	PORT_...	built i
GRAKAU...	10/28/98 13:51:45	IfUpDo...	void.sea...	ip.0	unreach...	Minor	PORT_...	built i
GRAKAU...	10/28/98 13:51:45	IfUpDo...	cobalt.s...	ip.0	unreach...	Minor	PORT_...	built i
GRAKAU...	10/28/98 13:51:45	IfUpDo...	mr-biggl...	ip.0	unreach...	Minor	PORT_...	built i
GRAKAU...	10/28/98 13:51:35	IfUpDo...	dallas.s...	ip.0	unreach...	Minor	PORT_...	built i
MOZART	10/28/98 15:01:56	IfDataL...	moneyp...	ifEntry.3	Logging	Normal	ifData	poll
MOZART	10/28/98 15:01:56	IfDataL...	moneyp...	ifEntry.2	Logging	Normal	ifData	poll
MOZART	10/28/98 15:01:56	IfDataL...	moneyp...	ifEntry.1	Logging	Normal	ifData	poll
MOZART	10/28/98 15:01:54	IfDataL...	ewing.s...	ifEntry.2	Logging	Normal	ifData	poll
MOZART	10/28/98 15:01:54	IfDataL...	ewing.s...	ifEntry.1	Logging	Normal	ifData	poll
MOZART	10/28/98 15:01:53	IfDatal...	[illegible]	ifEntry.2	Logging	Normal	ifData	poll

These windows are very similar. Both contain a tree view of the current alarm instances in the left pane, and details about the current alarm instances in the right pane. For information about how to interpret the information in these two panes, see the following sections:

- *The Tree View* on page 94
- *The Alarm-Detail View* on page 95

The Tree View

The left pane in both the Alarm Summary window and the Aggregate Alarm Summary window contains a tree of severities.



- RINGO 67
  - Severity 67
    - Fault 67
      - 🟥 Critical
      - 🟨 Major 5
      - 🟨 Minor 5
      - 🟦 Warning
      - 🟪 Inform 34
      - 🟪 Special
      - 🟩 Normal 23
    - Traffic
      - 🟪 Saturated
      - 🟦 VeryHigh
      - 🟦 High
      - ⬜ Medium
      - 🟩 Low
      - □ VeryLow

FIGURE 5-4. Severity Tree

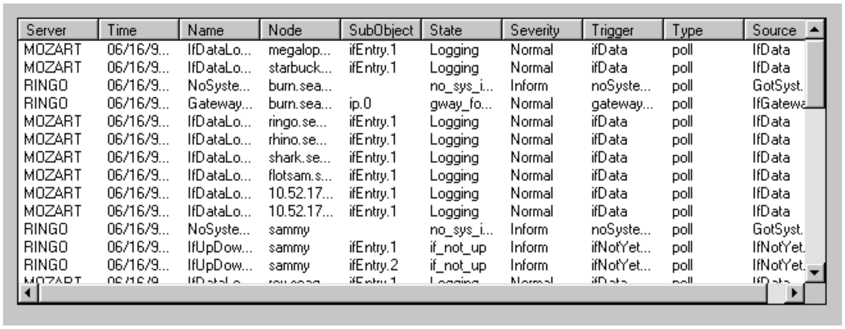
The only difference between the two trees is that the top folder in the Alarm Summary window represents the active server, while the top folder in the Aggregate Alarm Summary window represents all of the servers to which you are connected.

This tree view has two purposes:

- It enables you to see at a glance the total number of current alarm instances, the number of instances in each severity group (Fault and Traffic), and the number of instances of each severity. For instance, the tree above indicates that there are five alarm instances of Major severity.
- It enables you to control which alarm instances appear in the alarm detail pane. If you select the Major icon, only alarm instances of Major severity will appear in the alarm detail pane.

The Alarm-Detail View

The right pane in both the Alarm Summary and Aggregate Alarm Summary windows contains an alarm detail view that presents quite a bit of information about each current alarm instance, as shown in Figure 5-5.



Server	Time	Name	Node	SubObject	State	Severity	Trigger	Type	Source
MOZART	06/16/9...	IfDataLo...	megalop...	ifEntry.1	Logging	Normal	ifData	poll	IfData
MOZART	06/16/9...	IfDataLo...	starbuck...	ifEntry.1	Logging	Normal	ifData	poll	IfData
RINGO	06/16/9...	NoSyste...	burn.sea...		no_sys_i...	Inform	noSyste...	poll	GotSyst.
RINGO	06/16/9...	Gateway...	burn.sea...	ip.0	gway_fo...	Normal	gateway...	poll	IfGatewa
MOZART	06/16/9...	IfDataLo...	ringo.se...	ifEntry.1	Logging	Normal	ifData	poll	IfData
MOZART	06/16/9...	IfDataLo...	rhino.se...	ifEntry.1	Logging	Normal	ifData	poll	IfData
MOZART	06/16/9...	IfDataLo...	shark.se...	ifEntry.1	Logging	Normal	ifData	poll	IfData
MOZART	06/16/9...	IfDataLo...	flotsam.s...	ifEntry.1	Logging	Normal	ifData	poll	IfData
MOZART	06/16/9...	IfDataLo...	10.52.17...	ifEntry.1	Logging	Normal	ifData	poll	IfData
MOZART	06/16/9...	IfDataLo...	10.52.17...	ifEntry.1	Logging	Normal	ifData	poll	IfData
RINGO	06/16/9...	NoSyste...	sammy		no_sys_i...	Inform	noSyste...	poll	GotSyst.
RINGO	06/16/9...	IfUpDow...	sammy	ifEntry.1	if_not_up	Inform	ifNotYet...	poll	IfNotYet.
RINGO	06/16/9...	IfUpDow...	sammy	ifEntry.2	if_not_up	Inform	ifNotYet...	poll	IfNotYet.

FIGURE 5-5. Alarm Detail Pane

This is the pane you'll use for most of your monitoring.

Table 5-2 explains what information is available for each alarm instance.

TABLE 5-2. Fields in Alarm Detail Pane

Column	Description
Server	The name of the server that is managing the alarm instance. This column is present only in the Aggregate Alarm Summary window.
Time	The date and time at which the alarm instance's most recent transition occurred.
Name	The name of the alarm from which the alarm instance was created. If you have any question about what condition a particular alarm is monitoring, you can use the NerveCenter Client to view a definition of the alarm. For information about viewing such a definition, see the section *Getting Information about an Alarm* on page 98.
Node	The hostname of IP address of the node the alarm instance is monitoring.
SubObject	The subobject associated with the alarm instance. This subobject consists of a MIB base object plus an instance number, for example, ifEntry.1. The instance usually tells you which interface on a device is being monitored.
State	The current state of the alarm instance. The name of the state should indicate the condition NerveCenter is reporting. For example, if an instance of the alarm IfUpDownStatus is monitoring an interface and the current state of the alarm instance is "down," the operational status of the interface is down.
Severity	The severity of the alarm instance's current state.

TABLE 5-2. Fields in Alarm Detail Pane (Continued)

Column	Description
Trigger	The name of the trigger that caused the most recent alarm transition.
Type	The type of trigger that caused the most recent alarm transition. The possible types are poll, mask, fire (alarm), and built-in.
Source	The name of the poll, mask, or alarm that generated the trigger. Or, in the case of a built-in trigger, the name of the trigger. Given the name of the trigger that caused the transition and the name of the object that generated the trigger, you can pinpoint the exact cause of a transition. See the section *Getting Information about a Trigger* on page 100 for details on this subject.

The alarm detail pane is designed primarily for reading. However, there are a couple of actions you can take from this pane.

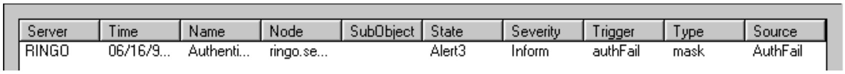
- You can select any of the column headings to sort the alarm-instance entries alphabetically by the values in that column. Selecting the column heading a second time reverses the order of the entries.

  This feature is useful for tasks such as ordering alarm instances by node.

- You can double-click the entry for an alarm instance to open an alarm-history window. For more information about alarm history, see the section *Viewing an Alarm Instance's History* on page 108.

Interpreting Alarm-Instance Information

The alarm-instance information that you can view using the NerveCenter Web Client and the NerveCenter Client is meant to stand on its own, that is, to provide you with all the information you need concerning a network condition. However, until you become familiar with all of the behavior models being used at your site, you might need some supplementary information. For example, suppose you see the summary information shown in Figure 5-6:



Server	Time	Name	Node	SubObject	State	Severity	Trigger	Type	Source
RINGO	06/16/9...	Authenti...	ringo.se...		Alert3	Inform	authFail	mask	AuthFail

FIGURE 5-6. Summary Alarm Information

It’s clear what node is being reported on. However, if you’re not familiar with the Authentication alarm, it may not be clear what it means for an instance of this alarm to be in the state Alert3. (As it turns out, this state indicates that a node has received three authentication-failure traps within a ten-minute period.) To find out what this state means, you can use the NerveCenter Client to look at the documentation for, and definition of, this alarm. For information on this subject, see the section *Getting Information about an Alarm* on page 98.

Also, it may not always be clear what condition caused the Source to generate the trigger that led to the most recent alarm transition. In the figure above, a mask called AuthFail generated the trigger authFail. You can probably guess that a trap mask responded to an authentication-failure trap. But what if you were monitoring an instance of the alarm ifErrorStatus (which monitors the percentage of error packets on an interface) and the poll MediumErrorRate fired the trigger mediumErrorRate. You could infer that NerveCenter had seen a moderate number of error packets on an interface, but what constitutes a medium error rate? You can find out by using the NerveCenter Client to read the documentation for, or definition of, the MediumErrorRate poll. For further information on interpreting the meaning of a trigger, see the section *Getting Information about a Trigger* on page 100.

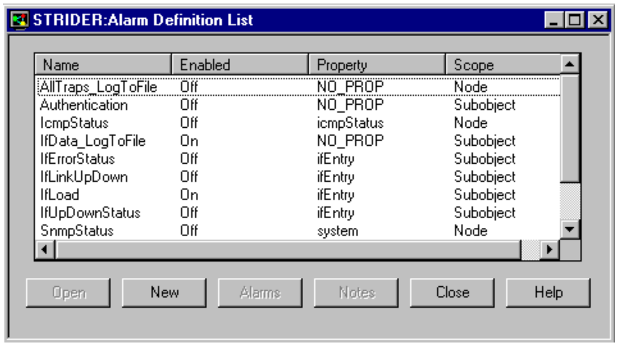
Getting Information about an Alarm

If you’re monitoring alarm instances and have a question about a particular alarm or alarm state, you can easily obtain information about the purpose of the alarm and what states are defined.

To get this information

1. From the NerveCenter Client’s **Admin** menu, choose **Alarm Definition List**.

   The Alarm Definition List window is displayed.



STRIDER:Alarm Definition List

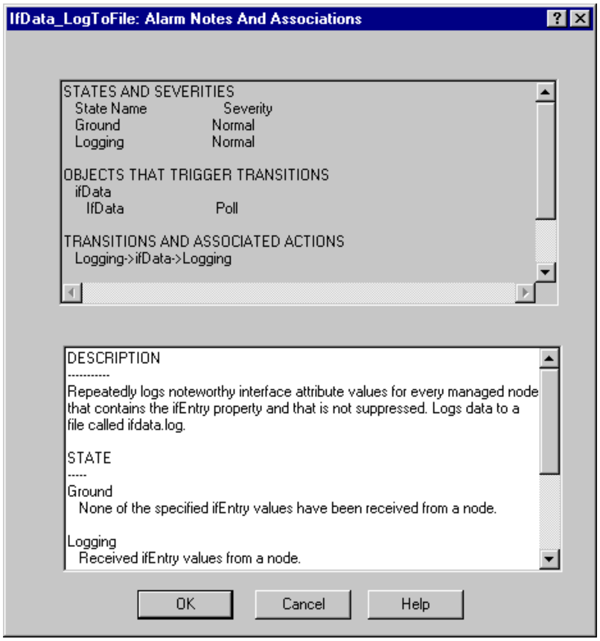
Name	Enabled	Property	Scope
AllTraps_LogToFile	Off	NO_PROP	Node
Authentication	Off	NO_PROP	Subobject
IcmpStatus	Off	icmpStatus	Node
IfData_LogToFile	On	NO_PROP	Subobject
IfErrorStatus	Off	ifEntry	Subobject
IfLinkUpDown	Off	ifEntry	Subobject
IfLoad	On	ifEntry	Subobject
IfUpDownStatus	Off	ifEntry	Subobject
SnmpStatus	Off	system	Node

2. Select the alarm you’re interested in.

   The **Notes** button is enabled.

3. Select the **Notes** button.

   The Alarm Notes and Associations dialog displays.



IfData_LogToFile: Alarm Notes And Associations

```
STATES AND SEVERITIES
  State Name              Severity
  Ground                Normal
  Logging               Normal

OBJECTS THAT TRIGGER TRANSITIONS
  ifData
    IfData                Poll

TRANSITIONS AND ASSOCIATED ACTIONS
  Logging->ifData->Logging
```

```
DESCRIPTION
-----------
Repeatedly logs noteworthy interface attribute values for every managed node
that contains the ifEntry property and that is not suppressed. Logs data to a
file called ifdata.log.

STATE
-----
Ground
  None of the specified ifEntry values have been received from a node.

Logging
  Received ifEntry values from a node.
```

These notes should include the following information:

- A list of states and their severities
- A list of transitions and the objects that can trigger those transitions
- A list of the actions associated with each transition
- A brief description of the purpose of the alarm
- A description of each state in the alarm
- Information about other alarms that are part of the same behavior model

NOTE

If you need further information about an alarm, you can look at its definition. To view this definition, select the alarm in the Alarm Definition List window and then select the **Open** button.

Getting Information about a Trigger

If you’re monitoring alarm instances and want further information about the cause of an alarm transition, you can obtain that information using the NerveCenter Client. The specific procedure you should follow depends on the type of the trigger that caused the transition. Table 5-3 directs you to the appropriate subsection.

TABLE 5-3. Finding Information about the Source of a Trigger

Type	See this Section
poll	*A Trigger Generated by a Poll* on page 100
mask	*A Trigger Generated by a Mask* on page 102
fire (alarm)	*A Trigger Generated by an Alarm* on page 104
built-in	*A List of Built-In Triggers* on page 104

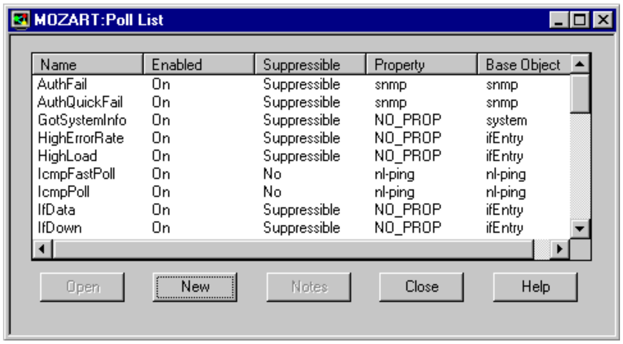
A Trigger Generated by a Poll

If you’re monitoring alarm instances and see that an alarm transition took place when the poll CsCpuBusy fired the csCpuBusy trigger, how can you determine what condition caused the poll to fire this trigger? To get this information, follow the procedure below.

TO DETERMINE WHY A POLL FIRED A TRIGGER

1. From the NerveCenter Client’s **Admin** menu, choose **Poll List**.

   The Poll List window is displayed.



MOZART:Poll List

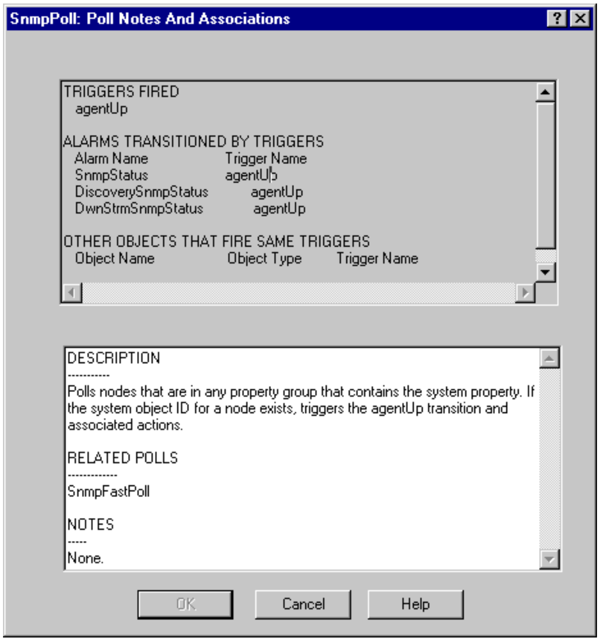
Name	Enabled	Suppressible	Property	Base Object
AuthFail	On	Suppressible	snmp	snmp
AuthQuickFail	On	Suppressible	snmp	snmp
GotSystemInfo	On	Suppressible	NO_PROP	system
HighErrorRate	On	Suppressible	NO_PROP	ifEntry
HighLoad	On	Suppressible	NO_PROP	ifEntry
IcmpFastPoll	On	No	nl-ping	nl-ping
IcmpPoll	On	No	nl-ping	nl-ping
IfData	On	Suppressible	NO_PROP	ifEntry
IfDown	On	Suppressible	NO_PROP	ifEntry

2. Select the poll you’re interested in.

The **Notes** button is enabled.

3. Select the **Notes** button.

The Poll Notes dialog displays.



```
TRIGGERS FIRED
  agentUp

ALARMS TRANSITIONED BY TRIGGERS
  Alarm Name              Trigger Name
  SnmpStatus              agentUp
  DiscoverySnmpStatus         agentUp
  DwnStrmSnmpStatus           agentUp

OTHER OBJECTS THAT FIRE SAME TRIGGERS
  Object Name             Object Type      Trigger Name
```

```
DESCRIPTION
-----------
Polls nodes that are in any property group that contains the system property. If
the system object ID for a node exists, triggers the agentUp transition and
associated actions.

RELATED POLLS
-------------
SnmpFastPoll

NOTES
-----
None.
```

For each poll, the note should include the following information:

- A list of the triggers the poll can fire
- A list of the alarms in which the poll can cause a transition
- A list of other objects that can fire any one of the triggers fired by this poll
- A brief description of the poll and its poll condition

NOTE

If you need additional information about a poll, you can look at its definition. To view this definition, select the poll in the Poll List window and then select the **Open** button.

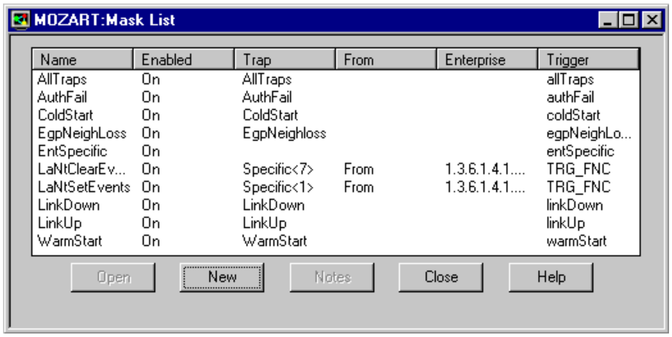
A Trigger Generated by a Mask

If you’re monitoring alarm instances and see that an alarm transition took place when the SynBoardPowerFail trap mask fired the synBoardPsTrap trigger, how can you determine what condition caused the mask to fire this trigger? To get this information, follow the procedure below.

To determine why a mask fired a trigger

1. From the NerveCenter Client’s **Admin** menu, choose **Mask List**.

   The Mask List window is displayed.



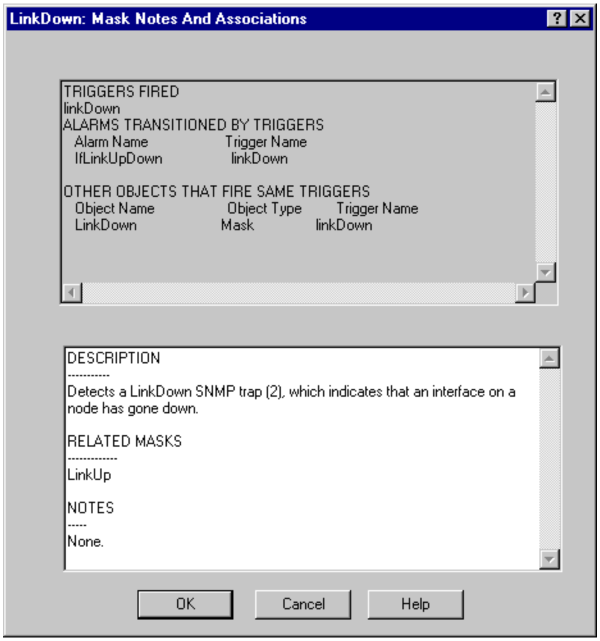
Name	Enabled	Trap	From	Enterprise	Trigger
AllTraps	On	AllTraps			allTraps
AuthFail	On	AuthFail			authFail
ColdStart	On	ColdStart			coldStart
EgpNeighLoss	On	EgpNeighloss			egpNeighLo...
EntSpecific	On				entSpecific
LaNtClearEv...	On	Specific<7>	From	1.3.6.1.4.1....	TRG_FNC
LaNtSetEvents	On	Specific<1>	From	1.3.6.1.4.1....	TRG_FNC
LinkDown	On	LinkDown			linkDown
LinkUp	On	LinkUp			linkUp
WarmStart	On	WarmStart			warmStart

2. Select the mask you’re interested in.

   The **Notes** button is enabled.

3. Select the **Notes** button.

   The Mask Notes dialog displays.



```
TRIGGERS FIRED
linkDown
ALARMS TRANSITIONED BY TRIGGERS
  Alarm Name                Trigger Name
  IfLinkUpDown              linkDown

OTHER OBJECTS THAT FIRE SAME TRIGGERS
  Object Name               Object Type     Trigger Name
  LinkDown                  Mask            linkDown
```

```
DESCRIPTION
-----------
Detects a LinkDown SNMP trap (2), which indicates that an interface on a
node has gone down.

RELATED MASKS
-------------
LinkUp

NOTES
-----
None.
```

For each mask, the note should include the following information:

- A list of the triggers that this mask can fire
- A list of the alarms in which this mask can cause a transition
- A list of other objects that fire any one of the triggers fired by this mask
- A brief description of the trap detected by this mask

NOTE

If you need additional information about a mask, you can look at its definition. To view this definition, select the mask in the Mask List window and then select the **Open** button.

A Trigger Generated by an Alarm

If you’re monitoring alarm instances and see that a transition in one alarm took place when another alarm fired a trigger, how can you determine what condition caused the Source alarm to fire this trigger? To get this information, you can look at the notes (documentation) for the Source alarm. Just follow the procedure mapped out in the section *Getting Information about an Alarm* on page 98.

A List of Built-In Triggers

If you’re monitoring alarm instances and an instance changes states because of a built-in trigger, you can consult the table below to determine why NerveCenter generated the built-in trigger. Table 5-4 lists all the built-in triggers that NerveCenter can fire.

NOTE
NerveCenter uses all uppercase letters to designate built-in trigger names.

TABLE 5-4. Built-In Triggers

Trigger Name	Meaning
CANNOT_SEND	A local error occurred while NerveCenter was trying to send an SNMP message.
ERROR	An SNMP or ICMP request did not result in a valid response. After firing the ERROR trigger, NerveCenter fires a second trigger that indicates the specific nature of the error.
ICMP_ERROR	Indicates an ICMP error. The ICMP_ERROR trigger contains the ICMP/IP fields from the error message.
ICMP_TIMEOUT	NerveCenter sent an ICMP ping to a node and did not receive a response. This trigger generally indicates that the node in question is down. NerveCenter uses the number of retries and retry interval specified on the SNMP tab in the Administrator. Refer to *Specifying SNMP Poll Intervals for NerveCenter* in *Managing NerveCenter* for details.
ICMP_UNKNOWN_ERROR	NerveCenter sent an ICMP ping to a node and received an invalid response. This trigger is no longer used except for the purpose of backward compatibility with version 3.5. We recommend you use it sparingly in the current version.
INFORM_CONNECTION_DOWN	A NerveCenter Inform host connection with OVPA or paserver is down.

TABLE 5-4. Built-In Triggers (Continued)

Trigger Name	Meaning
INFORM_CONNECTION_UP	A NerveCenter Inform host connection with OVPA or paserver was down but is now back up.
INFORMS_LOST	The number of NerveCenter Informs that were unacknowledged and lost, usually while the inform host connection with OVPA was down.
NET_UNREACHABLE	Indicates that the IP routing layer could not find a route to the network containing the polled node, usually because at least one router was down. This trigger indicates nothing about the status of the node. This trigger can be issued only if you have a router between the workstation running NerveCenter and the polled node.
NODE_UNREACHABLE	Indicates that the IP routing layer could not find a route to the destination node. This trigger indicates nothing about the status of the node. This trigger can be issued only if you have a router between the workstation running NerveCenter and the polled node.
PORT_UNREACHABLE	NerveCenter sent a message to a node, and there was no response from the port to which the message was sent.
RESPONSE	NerveCenter sent an SNMP message and received a valid response from the agent on the destination node.
SNMP_AUTHORIZATIONERR	An SNMP v3 authorization error caused because there is a mismatch between one or all of the rows of **vacmAccessTable** and the packet. Reasons include: context name mismatch (**vacmAccessContextPrefix**); security model is not used (**vacmAccessSecurityModel**); incorrect security level (**vacmAccessSecurityLevel**); unauthorized to read the MIB view for the SNMP context (**vacmAccessReadViewName**); unauthorized to write to the MIB view for the SNMP context (**vacmAccessWriteViewName**); unauthorized to notify the MIB view for the SNMP context (**vacmAccessNotifyViewName**)
SNMP_BADVALUE	NerveCenter tried to set the value of an attribute in a MIB, but the value it supplied was inappropriate for the attribute. The value may have been of the wrong type, of the wrong length, or invalid for some other reason.

TABLE 5-4. Built-In Triggers (Continued)

Trigger Name	Meaning
SNMP_DECRYPTION_ERROR	The SNMP v3 engine dropped packets because they could not be decrypted. The 32-bit counter, **usmStatsDecryptionErrors**, is greater than zero.
SNMP_ENDOFTABLE	NerveCenter fires **SNMP_ENDOFTABLE** when it finds no more rows while performing an SNMP walk of a MIB table. For example, you could walk **IfTable** to determine the number of DSO interfaces a node contains.
SNMP_GENERR	A GetRequest, GetNextRequest, or SetRequest failed for some unknown reason (general error).
SNMP_NOSUCHNAME	NerveCenter sent to an SNMP agent a GetRequest, a GetNextRequest, or a SetRequest, and the agent that was contacted was unable to perform the requested operation because: ◆ The name of the attribute to be read did not match exactly the name of an attribute available for get operations in the relevant MIB view ◆ The name of the attribute to be read did not lexicographically precede the name of an attribute available for get operations in the relevant MIB view ◆ The attribute to be set was not available for set operations in the relevant MIB view
SNMP_NOT_IN_TIME_WINDOW	The SNMP v3 engine dropped packets because the boots and timeticks sent in the PDU appeared outside of the authoritative SNMP agent's time window. The 32-bit counter, **usmStatsNotInTimeWindows**, is greater than zero.
SNMP_READONLY	The error readOnly is not defined in RFC 1157. However, some vendors’ agents do use this error-status code. As the name implies, the error usually indicates that an agent has received a SetRequest (from NerveCenter, in this case) for an attribute whose access type is read only.

TABLE 5-4. Built-In Triggers (Continued)

Trigger Name	Meaning
SNMP_TIMEOUT	NerveCenter sent an SNMP message to an agent and did not receive a response. This trigger indicates either that a node's SNMP agent is down or that the node itself is down. NerveCenter uses the number of retries and retry interval specified on the SNMP tab in the Administrator. Refer to t*Specifying SNMP Poll Intervals for NerveCenter* in *Managing NerveCenter* for details.
SNMP_TOOBIG	An SNMP agent did not respond normally to a GetRequest, GetNextRequest, or SetRequest from NerveCenter because the size of the required GetResponse would have exceeded a local limitation.
SNMP_UNAVAILABLE_CONTEXT	The SNMP v3 engine dropped packets because the context contained in the message was unavailable. The 32-bit counter, **snmpUnavailableContexts**, is greater than zero.
SNMP_UNKNOWN_CONTEXT	The SNMP v3 engine dropped packets because the context contained in the message was unknown. The 32-bit counter, **snmpUnknownContexts**, is greater than zero.
SNMP_UNKNOWN_ENGINEID	The SNMP v3 engine dropped packets because they referenced an **snmpEngineID** that was not known to the SNMP v3 engine. The 32-bit counter, **usmStatsUnknownEngineIDs**, is greater than zero.
SNMP_UNKNOWN_USERNAME	The SNMP v3 engine dropped packets because they referenced a user that was not known to the SNMP v3 engine. The 32-bit counter, **usmStatsUnknownUserNames**, is greater than zero.
SNMP_UNSUPPORTED_SEC_LEVEL	The SNMP v3 engine dropped packets because the requested security level is unknown or unavailable. The 32-bit counter, **usmStatsUnsupportedSecLevels**, is greater than zero.
SNMP_WRONG_DIGEST	The SNMP v3 engine dropped packets because they didn't contain the expected digest value. The 32-bit counter, **usmStatsWrongDigests**, is greater than zero.
UNKNOWN_ERROR	Some other error occurred.

One additional trigger, USER_RESET, is not available from the list of built-in triggers in NerveCenter. NerveCenter fires USER_RESET to trigger another state for an existing alarm instance when you reset the alarm instance using the right-click pop-up menu in the Alarm Summary or Aggregate Alarm Summary windows.

Viewing an Alarm Instance's History

Using the alarm-instance viewers provided by the NerveCenter Web Client and the NerveCenter Client, you can view all the current alarm instances for the servers to which you're connected. Sometimes, however, you also need historical information about an alarm instance. For example, let's say that a current alarm instance tells you that an interface on a router has been experiencing high traffic for the last ten minutes. You might also want to see whether this is a new problem or whether it has happened before. To get this information, you can ask to see the alarm instance's history. This history includes information about the alarm instance's twenty most recent transitions.

⚠ **CAUTION**

When an alarm instance returns to Ground state, it is deleted and no history for the instance is retained. To track a particular condition on a device or an interface across alarm instances, a behavior model must record data about alarm transitions in a log file or the Windows Event Log. For information on reading logs, see the section *Reading Logged Data* on page 114. For information on creating logs, see the manual *Alarm Actions* in *Designing and Managing Behavior Models*.

The procedure you use to view historical information depends on whether you're using the Web client or the Client to monitor your network. See one of the following subsections:

- *Using the NerveCenter Web Client* on page 109
- *Using the NerveCenter Client* on page 110

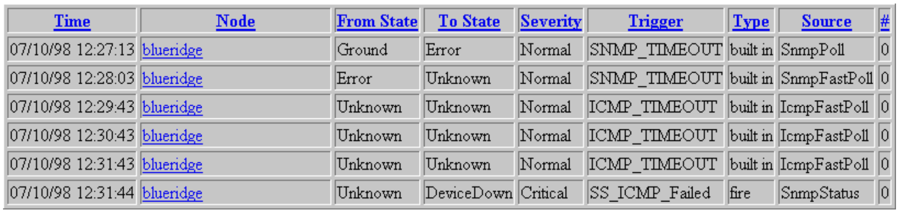
Using the NerveCenter Web Client

This section explains how to view the history of an alarm instance using the NerveCenter Web Client. To view this information, you go to the Web client's Alarm History page.

To go to the Alarm History page

- While on the alarm-summary page, select the **Name** field of the alarm instance whose history you want to see.

  This field is a hypertext link, and selecting it causes the Web client to display the Alarm History page.



Time	Node	From State	To State	Severity	Trigger	Type	Source	#
07/10/98 12:27:13	blueridge	Ground	Error	Normal	SNMP_TIMEOUT	built in	SnmpPoll	0
07/10/98 12:28:03	blueridge	Error	Unknown	Normal	SNMP_TIMEOUT	built in	SnmpFastPoll	0
07/10/98 12:29:43	blueridge	Unknown	Unknown	Normal	ICMP_TIMEOUT	built in	IcmpFastPoll	0
07/10/98 12:30:43	blueridge	Unknown	Unknown	Normal	ICMP_TIMEOUT	built in	IcmpFastPoll	0
07/10/98 12:31:43	blueridge	Unknown	Unknown	Normal	ICMP_TIMEOUT	built in	IcmpFastPoll	0
07/10/98 12:31:44	blueridge	Unknown	DeviceDown	Critical	SS_ICMP_Failed	fire	SnmpStatus	0

The Alarm History page displays the transitions that have led to the alarm instance's current state. The data displayed for each transition is similar to that displayed for an alarm instance on the alarm-summary page. The only new columns are From State, To State, and #. As you would guess, these columns hold the state of the alarm instance before the transition, the state of the instance after the transition, and the number of the transition (first, second, and so forth).

The figure above shows the history of an instance of a sample alarm, which monitors the status of a device and its SNMP agent. As you can see, the instance first transitioned from Ground to Error and then transitioned from Error to Unknown. Then, after receiving the built-in trigger ICMP_TIMEOUT several times, the instance transitioned to the Critical state DeviceDown.

There's only one action you can take from the Alarm History area. You can select the node field in the entry for any of the alarm instance's transitions to obtain information about the node associated with the alarm instance. For further information on this subject, see *To view information about a node* on page 140.

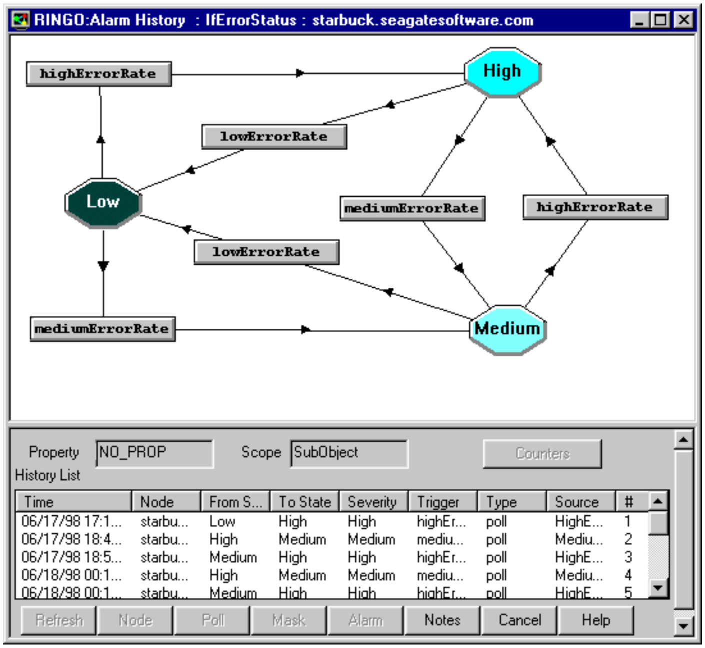
Using the NerveCenter Client

This section explains how to view the history of an alarm instance using the NerveCenter Client. To view this information, you bring up the Client’s Alarm History window.

To open the Alarm History window

- From the Alarm or Aggregate Alarm Summary window, double-click an alarm instance.

  The Alarm History window is displayed.



Property NO_PROP Scope SubObject

History List

Time	Node	From S...	To State	Severity	Trigger	Type	Source	#
06/17/98 17:1...	starbu...	Low	High	High	highEr...	poll	HighE...	1
06/17/98 18:4...	starbu...	High	Medium	Medium	mediu...	poll	Mediu...	2
06/17/98 18:5...	starbu...	Medium	High	High	highEr...	poll	HighE...	3
06/18/98 00:1...	starbu...	High	Medium	Medium	mediu...	poll	Mediu...	4
06/18/98 00:1...	starbu...	Medium	High	High	highEr...	poll	HighE...	5

The top pane in the Alarm History window displays the state diagram for the relevant alarm, and the list at the bottom of the window displays the transitions that have led to the alarm instance’s current state. The data displayed for each transition is similar to that displayed for an alarm instance in the Alarm or Aggregate Alarm Summary window. The only new columns are From State, To State, and #. These columns hold the state of the alarm instance before the transition, the state of the instance after the transition, and the number of the transition (first, second, and so forth).

The Alarm History window displays read-only information that cannot be modified. From this window, you can view the following:

TABLE 5-5. Alarm Instace History

Historical Item	Information Available
Alarm configuration	The alarm's property and scope appear in their respective fields. The unique identifier for the current alarm instance. Each alarm instance is assigned a unique instance ID by its associated NerveCenter Server.
Detail for each alarm instance	The list near the bottom of the window displays a line for each transition. Each line includes the following: ◆ Name of the trigger that caused the transition. ◆ Origin and destination states. ◆ Poll, mask or alarm that triggered the transition. ◆ Node whose agent caused the transition. ◆ Severity of the final state. ◆ Sequence of transitions for the instance. The 20 most recent transitions are listed in order of occurrence.
Transition activity	By selecting each entry in the list in order and watching the transitions change to red, you can follow along as the transitions lead to the current state of the alarm.
Transition configuration	Double-click the transition whose configuration you want to view. You can view the origin and destination states, the trigger that caused the transition, and any associated actions.
State severity	Double-click the state whose severity level you want to view.

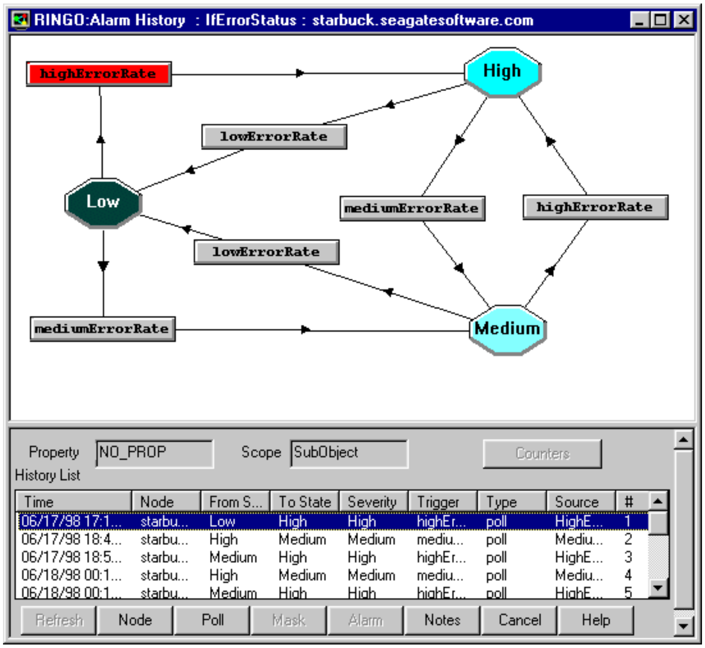
TABLE 5-5. Alarm Instace History (Continued)

Historical Item	Information Available
Associated NerveCenter objects	When you select an entry in the list, the **Node**, **Poll**, **Mask**, or **Alarm** buttons are enabled, as applicable, for each instance associated with one or more of these objects. Click a button to view the related node, poll, mask, or alarm definition. From a Node Definition window, you can reset alarms, ping a node, or obtain system information about the node.
Notes	Select the **Notes** button to view notes about the alarm.
Counters	Select the **Counters** button to see the names of any counters used in the alarm, along with the set value of each.

The preceding figure shows the history of an instance of the alarm IfErrorStatus, which monitors the percentage of error packets on an interface. As you can see, the instance first transitioned from low to high, and since then has bounced back and forth between the high and medium states. The times associated with the transitions (you can’t see all of them) indicate that the instance has been in the high state most of the time.

There are several actions that you can take from the Alarm History window:

- If you click the first transition in the alarm-history list, the corresponding transition in the state diagram is highlighted, as shown in Figure 5-7.



Property NO_PROP Scope SubObject

History List

Time	Node	From S...	To State	Severity	Trigger	Type	Source	#
06/17/98 17:1...	starbu...	Low	High	High	highEr...	poll	HighE...	1
06/17/98 18:4...	starbu...	High	Medium	Medium	mediu...	poll	Mediu...	2
06/17/98 18:5...	starbu...	Medium	High	High	highEr...	poll	HighE...	3
06/18/98 00:1...	starbu...	High	Medium	Medium	mediu...	poll	Mediu...	4
06/18/98 00:1...	starbu...	Medium	High	High	highEr...	poll	HighE...	5

FIGURE 5-7. Alarm History Window

By selecting the transitions in order—from first to last—you can watch the history of the alarm instance in the state-diagram pane.

- When you select a transition from the transition list, the **Node** button and typically either the **Poll**, **Mask**, or **Alarm** button are enabled. Selecting an enabled button opens a definition window that presents a definition of one of the following objects:
  - The node that the alarm instance is monitoring.
  - The poll that generated the trigger that caused the transition.
  - The trap mask that generated the trigger that caused the transition.
  - The alarm that generated the trigger that caused the transition.
- If the **Refresh** button becomes enabled while you’re viewing an alarm instance’s history, the alarm instance has undergone a transition while you’ve had the Alarm History window open. Select the **Refresh** button, and NerveCenter will update the information about this latest transition to the transition list.

Reading Logged Data

As was mentioned in the section *Viewing an Alarm Instance's History* on page 108, NerveCenter does not maintain a lot of historical information about network conditions. It remembers the last twenty transitions for each current alarm instance; however, when that alarm instance is deleted (when it returns to Ground), even that history is lost.

To preserve historical information about a network problem, a behavior model must log data about alarm transitions to a file, the system log (UNIX), Event Log (Windows), or to the NerveCenter database (Windows only). To take advantage of this logged data, all you need to know is where the data is being logged and how to interpret the logged data.

You can also manage the size of logs as well as the length of time they are stored by setting parameters in NerveCenter Administrator. For more information, see *Specifying Settings for Log Management* in *Managing NerveCenter*.

For more information about where NerveCenter writes log data and how you should interpret this data, see the following subsections:

- *Determining Where Data is Being Logged* on page 115
- *How to Interpret Logged Data* on page 117

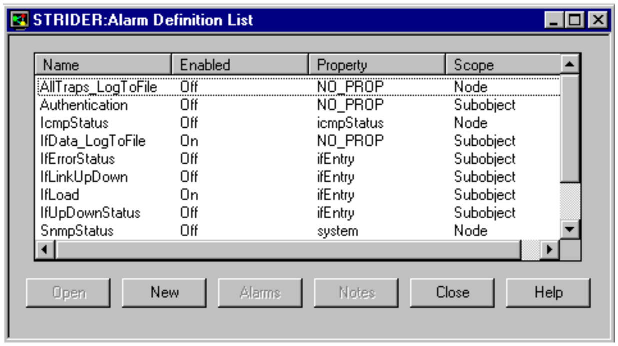
Determining Where Data is Being Logged

To determine whether an alarm logs data about any of its transitions and, if so, where it logs that data, you should look at the alarm's notes using the NerveCenter Client.

To view an alarm's notes

1. From the NerveCenter Client's **Admin** menu, choose **Alarm Definition List**.

   The Alarm Definition List window is displayed.



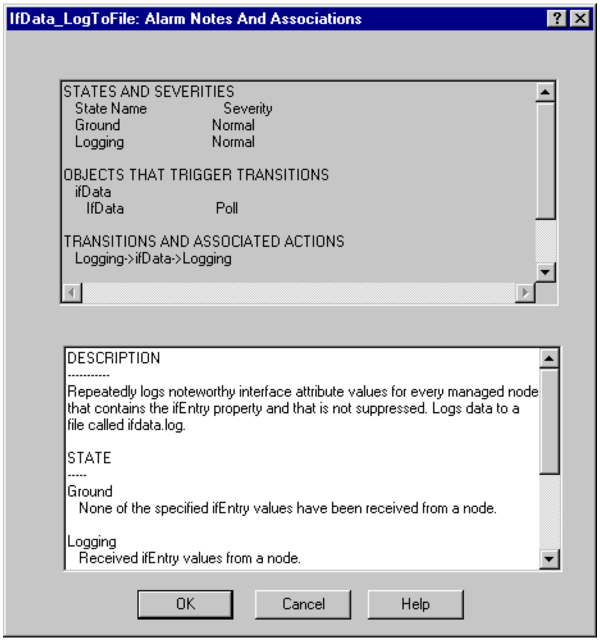
Name	Enabled	Property	Scope
AllTraps_LogToFile	Off	NO_PROP	Node
Authentication	Off	NO_PROP	Subobject
IcmpStatus	Off	icmpStatus	Node
IfData_LogToFile	On	NO_PROP	Subobject
IfErrorStatus	Off	ifEntry	Subobject
IfLinkUpDown	Off	ifEntry	Subobject
IfLoad	On	ifEntry	Subobject
IfUpDownStatus	Off	ifEntry	Subobject
SnmpStatus	Off	system	Node

2. Select the alarm you're interested in from the list of alarms.

   The **Notes** button is enabled.

3. Select the **Notes** button.

   The Alarm Notes and Associations dialog displays.



IfData_LogToFile: Alarm Notes And Associations

```
STATES AND SEVERITIES
  State Name            Severity
  Ground              Normal
  Logging             Normal

OBJECTS THAT TRIGGER TRANSITIONS
  ifData
    IfData              Poll

TRANSITIONS AND ASSOCIATED ACTIONS
  Logging->ifData->Logging
```

```
DESCRIPTION
-----------
Repeatedly logs noteworthy interface attribute values for every managed node
that contains the ifEntry property and that is not suppressed. Logs data to a
file called ifdata.log.

STATE
-----
Ground
  None of the specified ifEntry values have been received from a node.

Logging
  Received ifEntry values from a node.
```

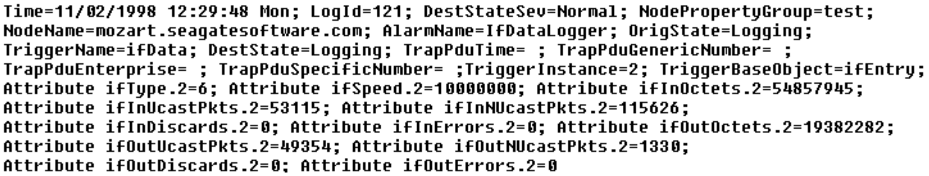
This dialog contains documentation for the alarm you selected and describes, among other things, any logging actions. For a Log to File action, the notes specify the log file to which data is written. An EventLog action causes NerveCenter to log data to one of the following locations:

- /var/adm/syslog/syslog.log (HP-UX)
- /var/adm/messages (Solaris)
- Windows Application event log

A Log to Database action causes NerveCenter to log to the NerveCenter database (Windows only).

How to Interpret Logged Data

Once you’ve determined where the data you’re interested in is being logged, you need to know how to interpret that data. Figure 5-8 shows a sample entry from a log file.



```
Time=11/02/1998 12:29:48 Mon; LogId=121; DestStateSev=Normal; NodePropertyGroup=test;
NodeName=mozart.seagatesoftware.com; AlarmName=IfDataLogger; OrigState=Logging;
TriggerName=ifData; DestState=Logging; TrapPduTime= ; TrapPduGenericNumber= ;
TrapPduEnterprise= ; TrapPduSpecificNumber= ;TriggerInstance=2; TriggerBaseObject=ifEntry;
Attribute ifType.2=6; Attribute ifSpeed.2=10000000; Attribute ifInOctets.2=54857945;
Attribute ifInUcastPkts.2=53115; Attribute ifInNUcastPkts.2=115626;
Attribute ifInDiscards.2=0; Attribute ifInErrors.2=0; Attribute ifOutOctets.2=19382282;
Attribute ifOutUcastPkts.2=49354; Attribute ifOutNUcastPkts.2=1330;
Attribute ifOutDiscards.2=0; Attribute ifOutErrors.2=0
```

FIGURE 5-8. Log File Entry

Table 5-6 explains what information the fields in this report contain.

TABLE 5-6. Fields in a Log Entry

Field	Contains
Time	Date and time the record was logged. The format of the time is *mm/dd/yyyy hh:mm:ss day* (for example, 10/29/1997 14:32:22 Sat).
LogID	Identification number of the log entry. NerveCenter assigns a sequential number to each log entry.
DestStateSev	Severity of the transition’s destination state.
NodeProperty	Property group of the node that caused the alarm to change states.
NodeName	Name of the node that caused the alarm to change states.
AlarmName	Name of the alarm.
OrigState	Name of the state from which the alarm moves when the logged transition occurs.
TriggerName	Name of the trigger that causes the alarm to move from the Ostate to the Nstate.
DestState	State of the alarm after the logged transition occurs.
TrapPduTime	The contents of a trap’s timestamp field. Used only when the transition is caused by a trap-mask trigger.
TrapPduGeneric	The contents of a trap’s generic-trap field. Used only when the transition is caused by a trap-mask trigger.
TrapPduEnterprise	The contents of a trap’s enterprise field. Used only when the transition is caused by a trap-mask trigger.
TrapPduSpecific	The contents of a trap’s specific-trap field. Used only when the transition is caused by a trap-mask trigger.

TABLE 5-6. Fields in a Log Entry (Continued)

Field	Contains
TriggerInstance	The specific base object instance for which the transition occurred.
TriggerBaseObject	The base object associated with the transition.
Attribute ...	The variable bindings of the trigger that caused the transition. Each variable binding is printed in the format Attribute *attribute.instance=value.*

NOTE

If a log file was created in non-verbose mode, its entries will not contain the labels shown in the figure above, but only a series of semi-colon-separated values.

When you're processing a log file created by a particular alarm, keep in mind the following rules:

- If the alarm has Enterprise scope, all the entries in the log constitute a single data set.
- If the alarm has Node scope, all entries that refer to the same node make up a data set.
- If the alarm has Subobject scope, the entries that share a node and subobject constitute a data set.
- If the alarm has Instance scope, the entries that share an instance of an alarm, regardless of the MIB objects listed, constitute a data set.
- A log that is the result of a poll transition contains only data pertinent to a poll. For example, TriggerBaseObject will contain a value, but TrapPduTime will not.
- A log that is the result of a trap mask transition contains only data pertinent to a trap. For example, TrapPduTime will contain a value, but TriggerBaseObject will not.

Resetting Alarms

6

Some alarms are designed to return to the Ground state when the condition they detect goes away. For example, the predefined alarm ifLoad tracks the level of traffic on an interface. As traffic increases, instances of this alarm may move from the Ground state to the medium state to the high state. Then, as traffic subsides, they may transition from the high state to the medium state to the Ground state. When these instances return to Ground state, they are automatically deleted from the Alarm Summary window.

Other alarms, however, are designed to remain in a terminal state until they are manually reset. For example, an instance of the predefined alarm Authentication transitions to the Intrusion state if a node receives four or more authentication-failure traps in a ten-minute period. The instance then remains in this state until it is manually reset.

Both the NerveCenter Web Client and the NerveCenter Client give you the ability to reset alarm instances, though in somewhat different ways. For information on the reset capabilities afforded by each client, see the following sections:

- *Using the NerveCenter Web Client* on page 120
- *Using the NerveCenter Client* on page 122

Whether resetting alarms from the Client or Web Client, if you reset an alarm to ground, any pending triggers fired by that alarm are cleared if the **Clear Triggers for Reset To Ground or Off** checkbox is checked in the Client’s alarms definition window for the alarm.

Using the NerveCenter Web Client

The NerveCenter Web Client enables you to reset a single alarm instance or all of the alarm instances listed in the alarm-summary window. The latter option gives you a good deal of flexibility since the alarm-summary window could contain all the alarm instances retrieved from a given NerveCenter server, the alarm instances of a given severity retrieved from a particular server, and so forth.

For more information on these two options for resetting alarms, see the following sections:

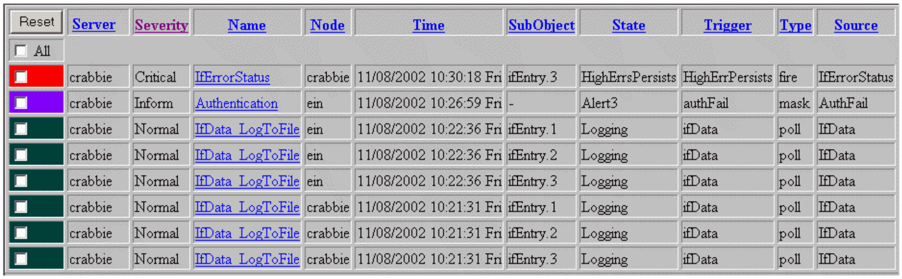
- *Resetting an Alarm Instance to Ground* on page 120
- *Resetting a Set of Alarms* on page 121

Resetting an Alarm Instance to Ground

To reset one or more alarm instances to Ground from the alarm-summary window, follow the directions below.

To Reset an Alarm Instance

1. Check the **Reset** checkbox for the alarm instance.



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

You can also check a number of checkboxes and reset all of the associated alarm instances of the checked item to Ground at once.

2. Select the **Reset** button at the top of the left column in the alarm-summary frame.

The alarm instance(s) is reset to Ground.

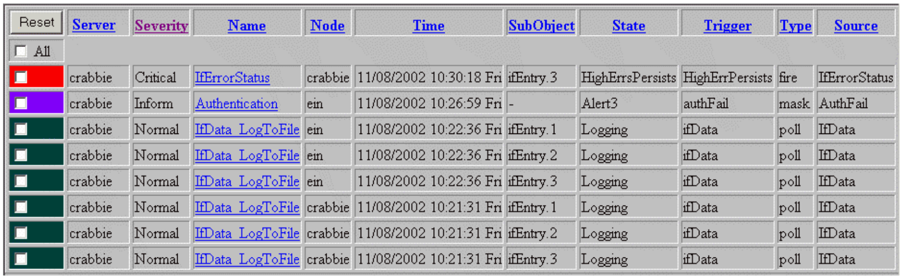
Resetting a Set of Alarms

As mentioned earlier, the NerveCenter Web Client enables you to reset, in one operation, all of the alarm instances listed in the alarm-summary window. Because the alarm-summary tree enables you to display information about a variety of sets of alarm instances, this feature is very flexible. For example, it enables you to reset:

- All the alarm instances retrieved from a particular server to ground
- All of the alarm instances of a given severity retrieved from a server to ground
- The alarm instances from one server that have a given severity and are monitoring nodes with a specific property group to ground
- All the alarm instances monitoring nodes on a particular partition to ground

To Reset a Set of Alarms

1. Display the alarm instances you want to reset by selecting the appropriate link in your alarm-summary tree.
2. In the alarm-detail frame, check the **All** checkbox located near the top left corner.



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

3. Select the **Reset** button in the upper left corner of the frame.

   The alarm instances are all reset to Ground.

Using the NerveCenter Client

The NerveCenter Client provides you with several ways to reset alarm instances. Most commonly, you'll select an alarm instance in the Alarm Summary or Aggregate Alarm Summary window and reset that instance to Ground. However, you can also select an instance in either of these windows and reset the state of the instance to any state allowed by the alarm. In addition, the NerveCenter Client enables you to reset either all alarm instances associated with a node or all alarm instances derived from the same alarm definition in a single operation.

For explanations of how to perform these reset operations, see the following sections:

- *Resetting an Alarm Instance to Ground* on page 123
- *Resetting an Alarm Instance to a Non-Ground State* on page 124
- *Resetting a Node's Alarm Instances* on page 125
- *Resetting All Instances of an Alarm* on page 127

Resetting an Alarm Instance to Ground

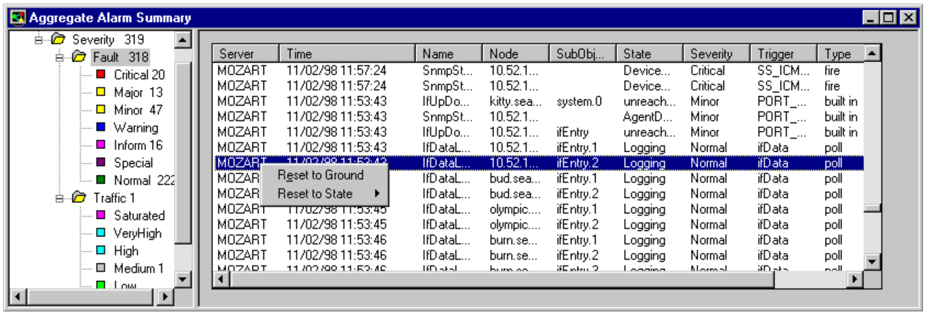
To reset one or more alarm instances to Ground from the Alarm or Aggregate Alarm Summary window, follow the directions below.

To reset an alarm instance

1. Select an alarm instance in the Alarm Summary or Aggregate Alarm Summary window.

   You can also select a number of alarm instances and reset all of them to Ground at once.

2. With your cursor positioned over the selected alarm instance, right-click to bring up the reset pop-up menu.



Aggregate Alarm Summary

Severity 319
Fault 318
Critical 20
Major 13
Minor 47
Warning
Inform 16
Special
Normal 222
Traffic 1
Saturated
VeryHigh
High
Medium 1
Low

Server	Time	Name	Node	SubObj...	State	Severity	Trigger	Type
MOZART	11/02/98 11:57:24	SnmpSt...	10.52.1...		Device...	Critical	SS_ICM...	fire
MOZART	11/02/98 11:57:24	SnmpSt...	10.52.1...		Device...	Critical	SS_ICM...	fire
MOZART	11/02/98 11:53:43	IfUpDo...	kitty.sea...	system.0	unreach...	Minor	PORT_...	built in
MOZART	11/02/98 11:53:43	SnmpSt...	10.52.1...		AgentD...	Minor	PORT_...	built in
MOZART	11/02/98 11:53:43	IfUpDo...	10.52.1...	ifEntry	unreach...	Minor	PORT_...	built in
MOZART	11/02/98 11:53:43	IfDataL...	10.52.1...	ifEntry.1	Logging	Normal	ifData	poll
MOZART		IfDataL...	10.52.1...	ifEntry.2	Logging	Normal	ifData	poll
MOZAR		IfDataL...	bud.sea...	ifEntry.1	Logging	Normal	ifData	poll
MOZAR		IfDataL...	bud.sea...	ifEntry.2	Logging	Normal	ifData	poll
MOZART		IfDataL...	olympic....	ifEntry.1	Logging	Normal	ifData	poll
MOZART	11/02/98 11:53:45	IfDataL...	olympic....	ifEntry.2	Logging	Normal	ifData	poll
MOZART	11/02/98 11:53:46	IfDataL...	burn.se...	ifEntry.1	Logging	Normal	ifData	poll
MOZART	11/02/98 11:53:46	IfDataL...	burn.se...	ifEntry.2	Logging	Normal	ifData	poll

3. Select **Reset to Ground** from the pop-up menu.

   The alarm instance reset to Ground and removed from the Alarm Summary list; however, if the network condition that caused that instance to be created in the first place still exists, a new alarm instance will be created to track that condition.

Note

In the Alarm Summary windows, if you select the instances for a single severity and change those instances to a state with a different, non-ground severity, they disappear from the current instance list. You can view them in the alarm list for their new severity.

If the **Clear Triggers for Reset To Ground or Off** checkbox is checked in the alarm's definition window, any pending triggers fired by that alarm are cleared when you reset the alarm to ground.

Resetting an Alarm Instance to a Non-Ground State

Sometimes you want to set an alarm instance to a state other than Ground. To reset one or more alarm instances to a state other than Ground from the Alarm or Aggregate Alarm Summary window, follow the directions below.

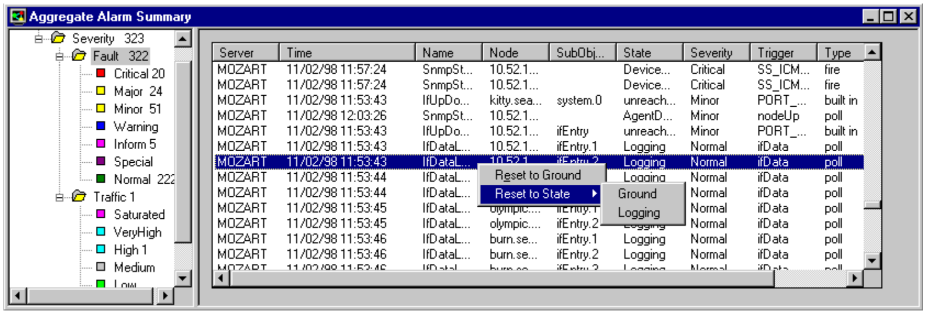
TO RESET AN ALARM INSTANCE

1. Select an alarm instance in the Alarm Summary or Aggregate Alarm Summary window.

   You can also select a number of alarm instances and reset all of them to a particular state at once. All of the instances must be derived from the same alarm definition.

2. With your cursor positioned over the selected alarm instance, right-click to bring up the reset pop-up menu.

3. Move your cursor over the **Reset to State** entry to bring up the state pop-up menu.



Server	Time	Name	Node	SubObj...	State	Severity	Trigger	Type
MOZART	11/02/98 11:57:24	SnmpSt...	10.52.1...		Device...	Critical	SS_ICM...	fire
MOZART	11/02/98 11:57:24	SnmpSt...	10.52.1...		Device...	Critical	SS_ICM...	fire
MOZART	11/02/98 11:53:43	IfUpDo...	kitty.sea...	system.0	unreach...	Minor	PORT_...	built in
MOZART	11/02/98 12:03:26	SnmpSt...	10.52.1...		AgentD...	Minor	nodeUp	poll
MOZART	11/02/98 11:53:43	IfUpDo...	10.52.1...	ifEntry	unreach...	Minor	PORT_...	built in
MOZART	11/02/98 11:53:43	IfDataL...	10.52.1...	ifEntry.1	Logging	Normal	ifData	poll
MOZART	11/02/98 11:53:43	IfDataL...	10.52.1	ifEntry.2	Logging	Normal	ifData	poll
MOZART	11/02/98 11:53:44	IfDataL...			Logging	Normal	ifData	poll
MOZART	11/02/98 11:53:44	IfDataL...				Normal	ifData	poll
MOZART	11/02/98 11:53:45	IfDataL...	olympic....	ifEntry.1		Normal	ifData	poll
MOZART	11/02/98 11:53:45	IfDataL...	olympic....	ifEntry.2	Logging	Normal	ifData	poll
MOZART	11/02/98 11:53:46	IfDataL...	burn.se...	ifEntry.1	Logging	Normal	ifData	poll
MOZART	11/02/98 11:53:46	IfDataL...	burn.se...	ifEntry.2	Logging	Normal	ifData	poll

4. Select the state to which you want to reset the instance from the pull-right menu.

   The entry for the alarm instance will show that it is now in the state you selected. The trigger that caused the transition to that state is the built-in trigger USER_RESET.

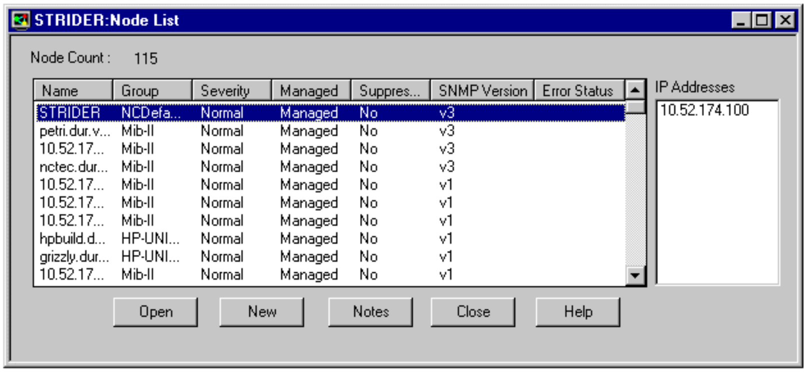
Resetting a Node’s Alarm Instances

Once you’ve identified a node that is experiencing a problem and have addressed the problem, you may want to reset all the alarm instances monitoring that node to Ground. You can reset all of these alarm instances using the procedure below:

TO RESET THE ALARM INSTANCES

1. From the client’s **Admin** menu, choose **Node List**.

   The Node List window is displayed.



STRIDER:Node List

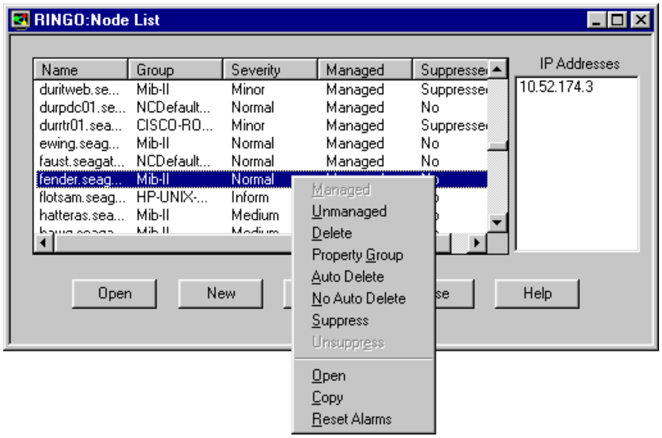
Node Count : 115

Name	Group	Severity	Managed	Suppres...	SNMP Version	Error Status
STRIDER	NCDefa...	Normal	Managed	No	v3	
petri.dur.v...	Mib-II	Normal	Managed	No	v3	
10.52.17...	Mib-II	Normal	Managed	No	v3	
nctec.dur...	Mib-II	Normal	Managed	No	v3	
10.52.17...	Mib-II	Normal	Managed	No	v1	
10.52.17...	Mib-II	Normal	Managed	No	v1	
10.52.17...	Mib-II	Normal	Managed	No	v1	
hpbuild.d...	HP-UNI...	Normal	Managed	No	v1	
grizzly.dur...	HP-UNI...	Normal	Managed	No	v1	
10.52.17...	Mib-II	Normal	Managed	No	v1	

IP Addresses
10.52.174.100

2. Select the node whose alarm instances you want to reset.

3. Place your cursor over the selected instance and right-click to bring up the node pop-up menu.



Name	Group	Severity	Managed	Suppresse
duritweb.se...	Mib-II	Minor	Managed	Suppresse
durpdc01.se...	NCDefault...	Normal	Managed	No
durrtr01.sea...	CISCO-RO...	Minor	Managed	Suppresse
ewing.seag...	Mib-II	Normal	Managed	No
faust.seagat...	NCDefault...	Normal	Managed	No
fender.seag...	Mib-II	Normal		
flotsam.seag...	HP-UNIX-...	Inform		
hatteras.sea...	Mib-II	Medium		

4. Select **Reset Alarms** from the pop-up menu.

NOTE

You can also reset all alarm instances associated with a node by opening the Node Definition window for a node, selecting the **Alarms** tab, and selecting the **Reset All** button.

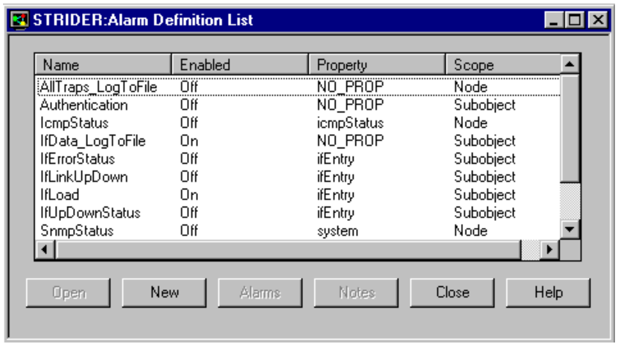
Resetting All Instances of an Alarm

When you select an alarm from the Alarm Definition List and perform a reset operation on it, you reset to Ground all the current instances of that alarm. That is, if you are using the Authentication behavior model and instances of the Authentication alarm have been instantiated for three nodes, all three instances will be deleted when you reset the Authentication alarm.

TO RESET AN ALARM

1. From the client’s **Admin** menu, choose **Alarm Definition List**.

   The Alarm Definition List window is displayed.

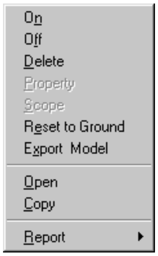


STRIDER:Alarm Definition List

Name	Enabled	Property	Scope
AllTraps_LogToFile	Off	NO_PROP	Node
Authentication	Off	NO_PROP	Subobject
IcmpStatus	Off	icmpStatus	Node
IfData_LogToFile	On	NO_PROP	Subobject
IfErrorStatus	Off	ifEntry	Subobject
IfLinkUpDown	Off	ifEntry	Subobject
IfLoad	On	ifEntry	Subobject
IfUpDownStatus	Off	ifEntry	Subobject
SnmpStatus	Off	system	Node

2. Select the alarm whose instances you want to reset from the list.

3. With your cursor over the selected alarm, right-click to bring up the alarm pop-up menu.



4. Select **Reset to Ground** from the pop-up menu.

6 Resetting Alarms

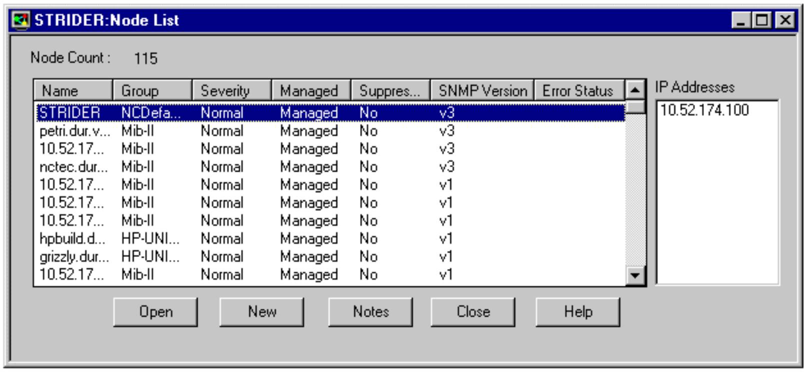
Monitoring SNMP Status and Operations

SNMP version 3 is an extension of SNMP that addresses security and administration. The following topics describe how NerveCenter provides support for SNMP v3. NerveCenter logs all SNMP operations to a file that you can use to track events and errors. In addition, the NerveCenter Client and Web Client provide error messages in their respective node lists for nodes with SNMP-related problems.

SNMP Error Status

When NerveCenter is unable to complete an SNMP operation on a node, the error status is displayed in the Node List (NerveCenter Client and Web Client) and in the SNMP tab of the node’s definition window (NerveCenter Client).

Figure 7-1 shows the Node List window in the Client.



Node Count : 115

Name	Group	Severity	Managed	Suppres...	SNMP Version	Error Status
STRIDER	NCDefa...	Normal	Managed	No	v3	
petri.dur.v...	Mib-II	Normal	Managed	No	v3	
10.52.17...	Mib-II	Normal	Managed	No	v3	
nctec.dur...	Mib-II	Normal	Managed	No	v3	
10.52.17...	Mib-II	Normal	Managed	No	v1	
10.52.17...	Mib-II	Normal	Managed	No	v1	
10.52.17...	Mib-II	Normal	Managed	No	v1	
hpbuild.d...	HP-UNI...	Normal	Managed	No	v1	
grizzly.dur...	HP-UNI...	Normal	Managed	No	v1	
10.52.17...	Mib-II	Normal	Managed	No	v1	

IP Addresses

10.52.174.100

FIGURE 7-1. 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

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.

NOTE

The dialogs are displayed only in NerveCenter Client, not in the NerveCenter Web Client.

When your NerveCenter Client or Administrator displays a dialog box with an error condition, you can do either of the following:

- Acknowledge the error condition by “signing the log.” When you sign the log, NerveCenter notes 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.

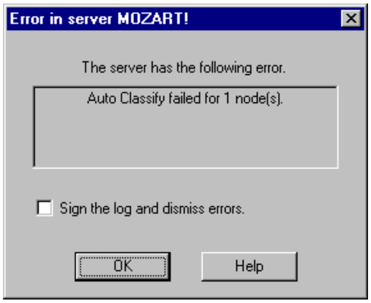
Whether you acknowledge or dismiss the error, all messages remain in the v3messages.log for you to read.

For more information, refer to the following topics:

- *Signing a Log for SNMP v3 Errors Associated with Your Client* on page 134
- *Signing a Log for SNMP v3 Errors Associated with a Remote Client or Administrator* on page 136
- *Viewing the SNMP v3 Operations Log* on page 137

Signing a Log for SNMP v3 Errors Associated with Your Client

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 Client that initiated the request causing a logged condition, NerveCenter displays a dialog box with the error that is logged as shown in Figure 7-2.


Error in server MOZART!

The server has the following error.

Auto Classify failed for 1 node(s).

□ Sign the log and dismiss errors.

FIGURE 7-2. Operations Log Error in Server Dialog Box for Your Client

Users with administrator rights can acknowledge a logged condition from NerveCenter Client by signing the Operations log. Signing the log causes the icon to turn green in all connected Clients/Administrators.

You can also dismiss the dialog box without acknowledging the error condition. If you are logged on with user rights rather than administrator rights, your only option is to dismiss the dialog box; you cannot sign the Operations log.

To Sign the Operations Log

1. After viewing the message that NerveCenter displays on your screen, check the Sign the log and dismiss errors checkbox.
2. 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 Client turns green. For all other connected Clients and Administrators, the icon remains red and signals to those modules that the NerveCenter Server has some error that remains unacknowledged.

Signing a Log for 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 Client (one that did not initiate the error condition), you see a red icon in the status bar.

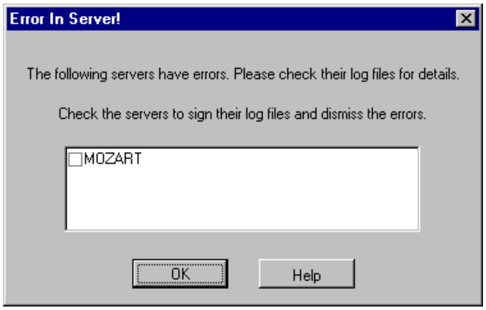
Users with administrator rights can acknowledge a logged condition from NerveCenter Client by signing the Operations log. Signing the log causes the icon to turn green in all connected Clients/Administrators.

You can also dismiss the dialog box without acknowledging the error condition. If you are logged on with user rights rather than administrator rights, your only option is to dismiss the dialog box; you cannot sign the Operations log.

To sign the Operations log

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

   The Error In Server dialog box is displayed.



Error In Server!

The following servers have errors. Please check their log files for details.

Check the servers to sign their log files and dismiss the errors.

□ MOZART

2. Check the NerveCenter Server or Servers for which you want to sign the log.
3. 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 Client turns green. For all other connected Clients and Administrators, the icon remains red and signals to those modules that the NerveCenter Server has some error that remains unacknowledged.

Viewing the 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;
```

Table 7-1 describes the fields in the log:

TABLE 7-1. 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 130 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.

7 Monitoring SNMP Status and Operations

Monitoring Nodes

8

While the principal way of viewing information in the NerveCenter Web Client and the NerveCenter Client is to view sets of alarm instances, both clients also enable you to monitor (or retrieve information about) an individual managed node. Because you'll probably use one client or the other for the majority of your monitoring, this chapter is divided into two main parts: one part covering the NerveCenter Web Client and the other covering the NerveCenter Client.

Section	Description
Using the NerveCenter Web Client on page 140	Explains how to use the NerveCenter Web Client to obtain information about a node.
Using the NerveCenter Client on page 142	Explains how to use the NerveCenter Client to monitor individual nodes.

Using the NerveCenter Web Client

The NerveCenter Web Client doesn’t have the node *monitoring* capabilities of the NerveCenter Client; however, it enables you to see the definition of the node associated with a particular alarm instance. The information that makes up this definition is pretty much the same information you can get from the Node Definition window in the NerveCenter Client.

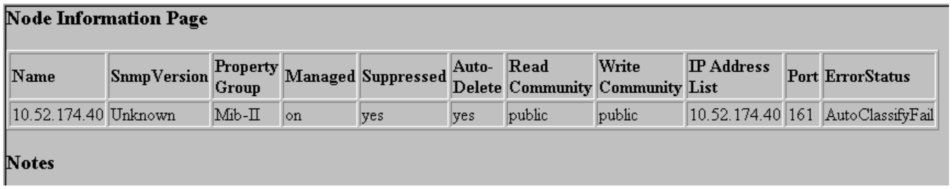
To view information about a node

1. While viewing the alarm-summary page, select the **Name** field of the alarm instance in which you’re interested.

   The Web client displays the Alarm History page for the selected alarm instance.

2. In the Alarm History table, select any of the hypertext links in the **Node** column.

   The Web client displays the Node Information page, which includes a table of data about the node associated with the alarm instance and any notes associated with the node.



Node Information Page

Name	SnmpVersion	Property Group	Managed	Suppressed	Auto-Delete	Read Community	Write Community	IP Address List	Port	ErrorStatus
10.52.174.40	Unknown	Mib-II	on	yes	yes	public	public	10.52.174.40	161	AutoClassifyFail

Notes

Table 8-1 explains how to interpret the data in this table.

TABLE 8-1. Definitions of Node Attributes

Data Member	Definition
Name	Contains the name of the workstation or network device being monitored. The name can be a hostname or an IP address.
SNMP Version	Indicates whether the node has been configured with an agent for SNMP version one, two, or three, or if the version is unknown. NerveCenter doesn’t poll nodes with an unknown version.
Property Group	Contains the node’s property group. This property group helps determine whether a particular poll can query this node and whether a particular alarm can be instantiated for the node.

TABLE 8-1. Definitions of Node Attributes (Continued)

Data Member	Definition
Managed	On or off. Indicates whether the node is to be managed by NerveCenter or not. By default, all nodes discovered by NerveCenter or a network management platform are managed.
Suppressed	Yes or no. Indicates whether the node is in a suppressed state. Suppressing a node limits polling because if the node is suppressed and a related poll is suppressible, that poll cannot cause an SNMP GetRequest to be sent to the node.
Auto Delete	Yes or no. Used when NerveCenter is integrated with a network management platform. If a node is removed from the platform's database, NerveCenter removes the node from its database if this attribute is yes.
Read Community	Contains the community name that NerveCenter will include in any SNMP GetRequest or GetNextRequest that it sends to the agent on this node. By default, set to public.
Write Community	Contains the community name that NerveCenter will include in any SNMP SetRequest that it sends to the agent on this node. By default, set to public.
IP Address List	Contains the node's IP address. If the node is multihomed, IP Address List can contain a list of addresses.
Port	Contains the number of the port that the node's agent uses to receive SNMP messages. By default, the port is set to 161.
Error Status	Lists the current SNMP error status, if applicable, for a node. Polls will not happen for any nodes whose error status is one of the following: ◆ AuthKeyFail ◆ PrivKeyFail ◆ AuthPrivKeyFail ◆ V3InitFail ◆ TestVersionFail ◆ ClassifyFail See *SNMP Error Status* on page 130 for a full list of SNMP errors.

Using the NerveCenter Client

The NerveCenter Client enables you to perform two types of node-related tasks. First, it enables you to view a list of alarms that can be instantiated for a node and a list of the alarms that are currently instantiated for a node. Second, the NerveCenter Client enables you to quickly query a node to determine its status.

For further information about these subjects, see the following subsections:

- *Viewing Related Alarms* on page 142
- *Querying Nodes* on page 145
- *Viewing Parent Node Status* on page 148

Viewing Related Alarms

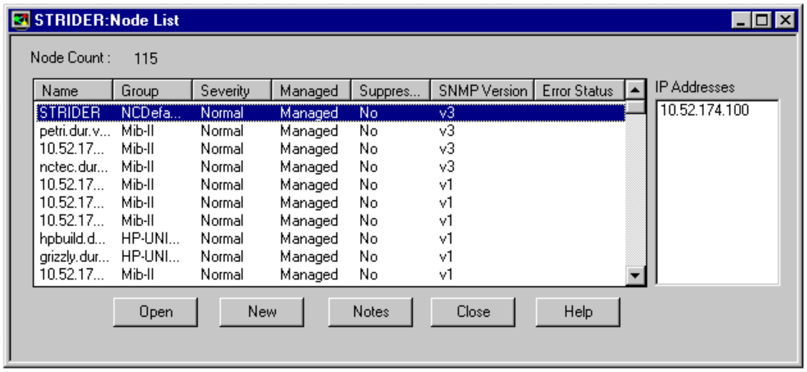
The NerveCenter Client can provide you with lists of:

- The alarms that can be instantiated for a node.
- The alarms that *have been* instantiated for a node. For each alarm instance, NerveCenter lists an alarm name, the enabled status of the alarm, the alarm's state, the subobject the alarm is monitoring, and the time at which the alarm instance was created. This information enables you to monitor the status of a node from the Node Definition window instead of the Alarm or Aggregate Alarm Summary window.

TO VIEW THE ALARMS RELATED TO A NODE

1. From the Client's **Admin** menu, choose **Node List**.

   The Node List window is displayed.



STRIDER:Node List

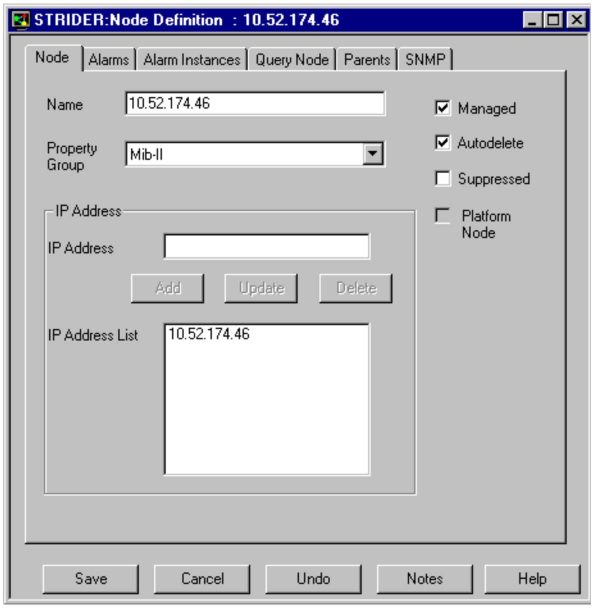
Node Count : 115

Name	Group	Severity	Managed	Suppres...	SNMP Version	Error Status
STRIDER	NCDefa...	Normal	Managed	No	v3	
petri.dur.v...	Mib-II	Normal	Managed	No	v3	
10.52.17...	Mib-II	Normal	Managed	No	v3	
nctec.dur...	Mib-II	Normal	Managed	No	v3	
10.52.17...	Mib-II	Normal	Managed	No	v1	
10.52.17...	Mib-II	Normal	Managed	No	v1	
10.52.17...	Mib-II	Normal	Managed	No	v1	
hpbuild.d...	HP-UNI...	Normal	Managed	No	v1	
grizzly.dur...	HP-UNI...	Normal	Managed	No	v1	
10.52.17...	Mib-II	Normal	Managed	No	v1	

IP Addresses: 10.52.174.100

2. Double-click the name or IP address of the node in which you're interested.

   The Node Definition window is displayed and presents the definition of the node you selected.



STRIDER:Node Definition : 10.52.174.46

Name: 10.52.174.46

Property Group: Mib-II

☑ Managed

☑ Autodelete

☐ Suppressed

☐ Platform Node

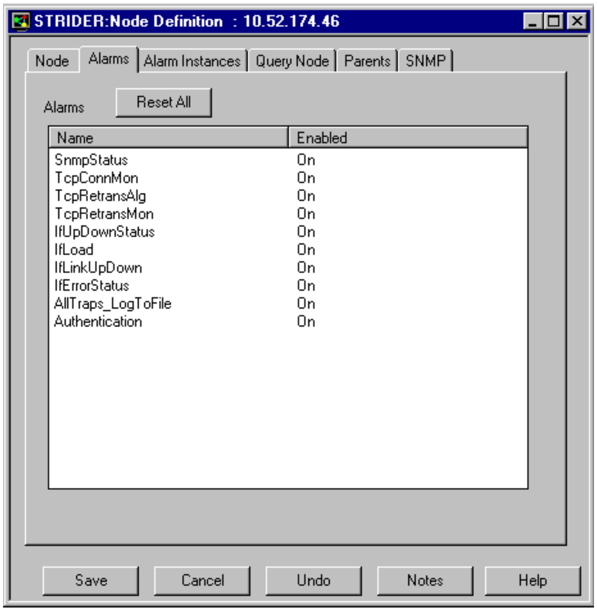
IP Address

IP Address:

IP Address List: 10.52.174.46

3. Select the **Alarms** tab.

The Alarms tab is displayed.



Name	Enabled
SnmpStatus	On
TcpConnMon	On
TcpRetransAlg	On
TcpRetransMon	On
IfUpDownStatus	On
IfLoad	On
IfLinkUpDown	On
IfErrorStatus	On
AllTraps_LogToFile	On
Authentication	On

In this example, the only alarm that has been instantiated is IfLoad. This alarm instance is monitoring interface 2 and is in the state medium, indicating that there is a moderate level of traffic on this interface.

The other alarms in the list are alarms that will be instantiated if:

- All the necessary NerveCenter objects are enabled
- The conditions that the alarms are designed to monitor actually occur

To see the documentation for an alarm, double-click the entry for an alarm to bring up the Alarm History window; then, click the **Notes** button.

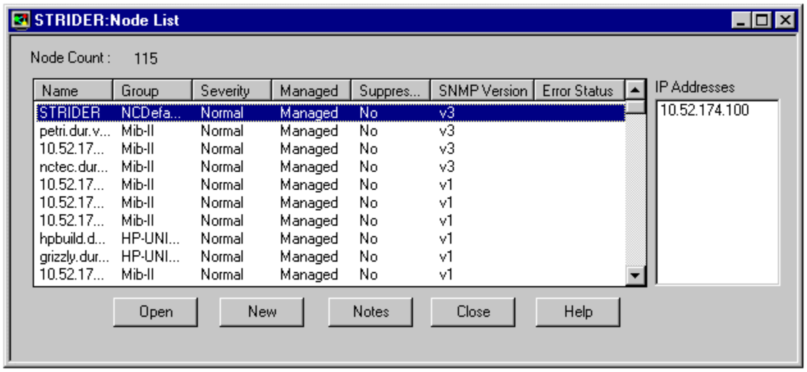
Querying Nodes

In addition to listing a node’s current alarm instances, the NerveCenter Client can query a node to determine whether the node is up and whether its SNMP agent is up—without using a behavior model. To determine whether a node is up or down, you send an ICMP ping to the node, and to determine the status of a node’s SNMP agent, you send an SNMP GetRequest to the node asking for information about the system object.

TO QUERY A NODE

1. From the Client’s **Admin** menu, choose **Node List**.

   The Node List window is displayed.



STRIDER:Node List

Node Count : 115

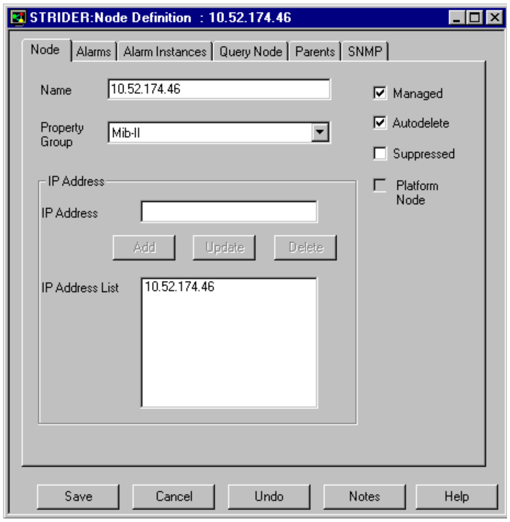
Name	Group	Severity	Managed	Suppres...	SNMP Version	Error Status
STRIDER	NCDefa...	Normal	Managed	No	v3	
petri.dur.v...	Mib-II	Normal	Managed	No	v3	
10.52.17...	Mib-II	Normal	Managed	No	v3	
nctec.dur...	Mib-II	Normal	Managed	No	v3	
10.52.17...	Mib-II	Normal	Managed	No	v1	
10.52.17...	Mib-II	Normal	Managed	No	v1	
10.52.17...	Mib-II	Normal	Managed	No	v1	
hpbuild.d...	HP-UNI...	Normal	Managed	No	v1	
grizzly.dur...	HP-UNI...	Normal	Managed	No	v1	
10.52.17...	Mib-II	Normal	Managed	No	v1	

IP Addresses

10.52.174.100

2. Double-click the name or IP address of the node in which you’re interested.

   The Node Definition window is displayed and presents the definition of the node you selected.



STRIDER:Node Definition : 10.52.174.46

Name: 10.52.174.46

Property Group: Mib-II

☑ Managed

☑ Autodelete

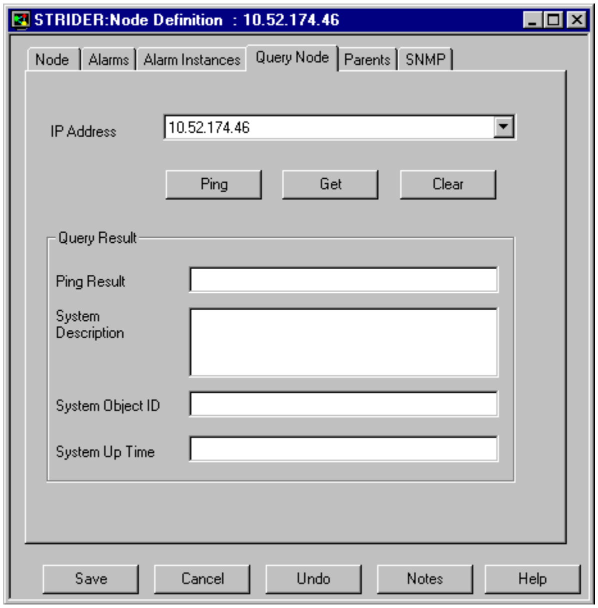
□ Suppressed

□ Platform Node

IP Address List: 10.52.174.46

3. Select the **Query Node** tab.

   The Query Node tab is displayed.



STRIDER:Node Definition : 10.52.174.46

IP Address: 10.52.174.46

4. If the node you’re querying is multihomed, select the IP address you want to use for the query from the **IP Address** drop-down list box.

5. Select the **Ping** or **Get** button.

   If you select the **Ping** button, NerveCenter pings the node, and the node’s response is displayed in the **Ping Result** field. If there is no response, the node is unreachable.

   If you select the **Get** button, NerveCenter sends to the node an SNMP GetRequest asking for the values of the following MIB attributes: sysDescr, sysObjectID, and sysUpTime. If the node replies with a GetResponse message that contains the values of these attributes, NerveCenter displays the values in the **System Description**, **System Object ID**, and **System Up Time** fields. If the node does not respond, you can infer that the node’s SNMP agent is down or that the node is unreachable.

NOTE

NerveCenter must know the node’s SNMP version before it can perform a GetRequest message.

Viewing Parent Node Status

NerveCenter monitors parent-child relationships and uses this information for *downstream alarm suppression*, suppressing alarms from any nodes that are downstream from a down router. The **Parents** tab of the Node Definition window displays the status that is obtained from NerveCenter's SetNodeStatus Perl subroutines.

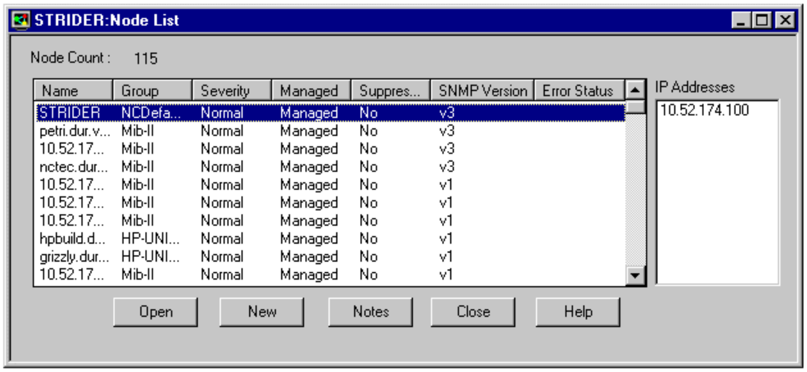
NOTE

The NerveCenter downstream alarm suppression alarms must be turned on before you can monitor a node's parents. Refer *Downstream Alarm Suppression* in *Designing and Managing Behavior Models*.

TO VIEW THE STATUS OF PARENT NODES

1. From the Client's **Admin** menu, choose **Node List**.

   The Node List window is displayed.



Node Count : 115

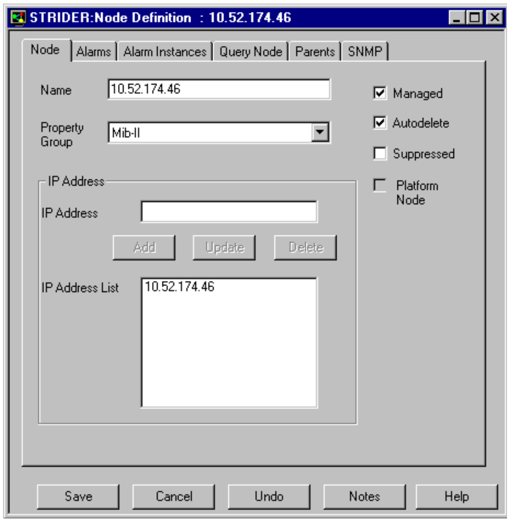
Name	Group	Severity	Managed	Suppres...	SNMP Version	Error Status
STRIDER	NCDefa...	Normal	Managed	No	v3	
petri.dur.v...	Mib-II	Normal	Managed	No	v3	
10.52.17...	Mib-II	Normal	Managed	No	v3	
nctec.dur...	Mib-II	Normal	Managed	No	v3	
10.52.17...	Mib-II	Normal	Managed	No	v1	
10.52.17...	Mib-II	Normal	Managed	No	v1	
10.52.17...	Mib-II	Normal	Managed	No	v1	
hpbuild.d...	HP-UNI...	Normal	Managed	No	v1	
grizzly.dur...	HP-UNI...	Normal	Managed	No	v1	
10.52.17...	Mib-II	Normal	Managed	No	v1	

IP Addresses

10.52.174.100

2. Double-click the name or IP address of the node in which you're interested.

   The Node Definition window is displayed and presents the definition of the node you selected.



STRIDER:Node Definition : 10.52.174.46

Name 10.52.174.46

Property Group Mib-II

☑ Managed

☑ Autodelete

□ Suppressed

□ Platform Node

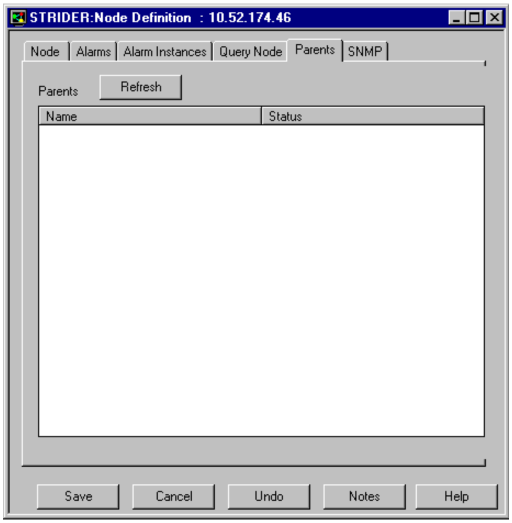
IP Address

IP Address

IP Address List 10.52.174.46

3. Select the **Parents** tab.

   The Parents tab is displayed.



STRIDER:Node Definition : 10.52.174.46

Parents

Name	Status

4. Select **Refresh** to update parent status information displayed in the window.

Generating Reports

9

As you work with NerveCenter, you may want to generate reports that describe NerveCenter objects or the activity on your network.

For further information about generating reports, see the appropriate section of this chapter.

Section	Description
Reports Shipped with NerveCenter on page 152	Describes the reports shipped with NerveCenter for the Windows environment.
Generating a Report on page 155	Describes how to run reports from the NerveCenter Reports window.

Reports Shipped with NerveCenter

NerveCenter ships three reports designed to report availability and outage levels by property group. These reports are to be used with the logging versions of the downstream alarm suppression behavior model. The reports are:

- **availsum**—Summarizes the availability and outage levels for each device by property group
- **availstat**—Provides actual and percentage values for the amount of time each device has been in a particular state (Ground or DeviceDown, for example)
- **availtrans**—Provides a detailed list of every transition for every device for the specified period of time

NerveCenter also ships another report (eventlog) that provides Windows Event Log data from the Application Log on the NerveCenter Server concerning NerveCenter events.

See *Downstream Alarm Suppression* in *Designing and Managing Behavior Models* for more details about the behavior model and these related reports.

For instructions, see *Generating a Report* on page 155.

Adding a Report

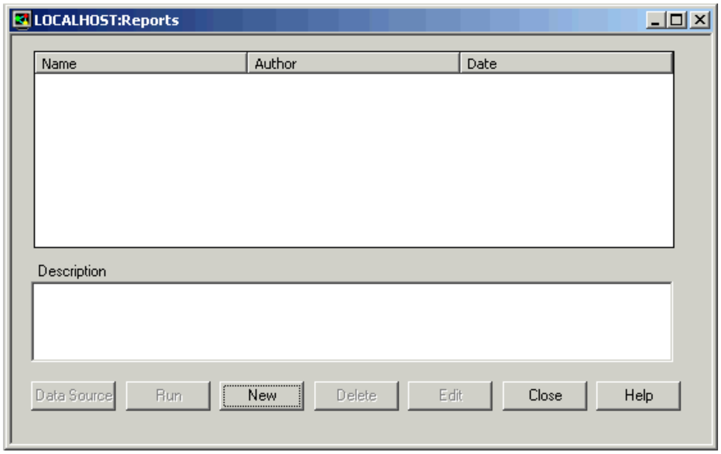
NOTE

Before you can add a custom report to the NerveCenter Client, the custom report file must reside in CustomRPT, found under the NerveCenter installation folder.

TO ADD A CUSTOM REPORT

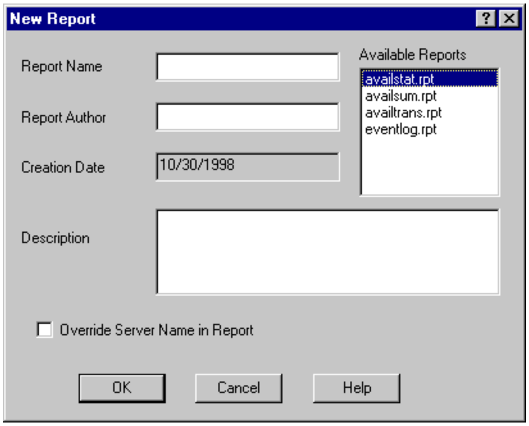
1. Choose **Reports** from the client's **Admin** menu.

   The Reports window is displayed.



2. Select the **New** button.

   The New Report dialog is displayed.



New Report

Report Name

Report Author

Creation Date 10/30/1998

Available Reports

availstat.rpt
availsum.rpt
availtrans.rpt
eventlog.rpt

Description

□ Override Server Name in Report

3. Enter the name for the report in the **Report Name** field.

4. Select the file you want to use as your report from the **Available Reports** list. This list contains the names of available Crystal Reports files extracted from the CustomRPT folder on the server.

5. You can optionally enter the name of the author and change the date. Once the report is added to the Report List window, you can sort and search reports by either field.

6. In the **Description** textbox, enter the description you want to appear on the Report List window.

7. If the report you are adding is an ODBC-based report (such as the downstream node availability reports shipped with NerveCenter), select the check box for **Override Server Name in Report**. This check box forces NerveCenter to run the report against the NerveCenter database on the NerveCenter Server rather than the database server. (This feature enables you to construct reports locally before you run them on the system on which NerveCenter is installed.)

   If the report is based on an event log, clicking this check box causes the report to run against the event log on the NerveCenter Server. Otherwise, the report runs against the event log on the NerveCenter Client.

8. Select **OK**.

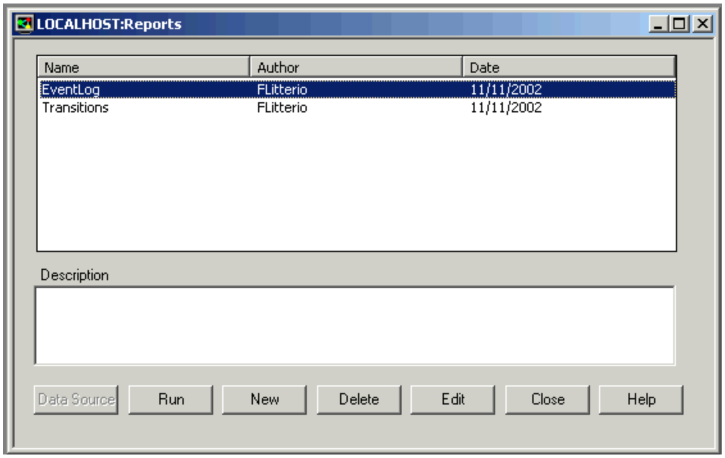
   The new report is added to the Report List window.

Generating a Report

TO GENERATE A REPORT

1. Choose **Reports** from the client's **Admin** menu.

   The Reports window is displayed.



LOCALHOST:Reports

Name	Author	Date
EventLog	FLitterio	11/11/2002
Transitions	FLitterio	11/11/2002

2. Select the appropriate report from the report list.
3. Select the **Run** button.

   The report you requested is displayed.

Using the Report Window's Controls

All NerveCenter reports presented on Windows systems appear in a window that has the toolbar shown in Figure 9-1.

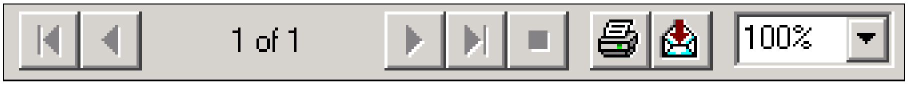


FIGURE 9-1. Report Window's Toolbar

Table 9-1 explains what the buttons in the report window's tool bar do.

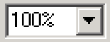
TABLE 9-1. Report Window Buttons

Button	What it does
⏮	Takes you to the beginning of the report.
◀	Takes you to the previous page in the report.
▶	Takes you to the next page in the report.
⏭	Takes you to the last page in the report.
🖨	Enables you to print the report to your default printer. You can print the entire report or selected pages, and you can print one copy or multiple copies.
✉	Enables you to export your report in one of many formats to one of several destinations. These formats and destinations are listed below.

Formats:

- Character-separated values
- Comma-separated values (CSV)
- Crystal Reports (RPT
- Data Interchange Format (DIF)
- Excel 2.1 (XLS)
- Excel 3.0 (XLS)
- Excel 4.0 (XLS)
- Excel 5.0 (XLS)
- Excel 5.0 (XLS) Tabular
- HTML 3.0 (Draft Standard)
- HTML 3.2 (Extended)
- HTML 3.2 (Standard)
- Lotus 1-2-3
- Lotus 1-2-3 (WK1)
- Lotus 1-2-3 (WK3)
- Lotus 1-2-3 (WKS)
- ODBC - dBASE Files
- ODBC - Excel Files
- ODBC - FoxPro Files
- ODBC - MS Access 97 Database
- ODBC - NC35
- ODBC - Text Files
- Paginated text
- Record style (columns of values)
- Rich Text Format
- Tab-separated text
- Tab-separated values
- Text
- Word for Windows document

TABLE 9-1. Report Window Buttons (Continued)

Button	What it does
	Destinations: ◆ Disk file ◆ Exchange folder ◆ Microsoft Mail
	Allows the adjustment of the view of the report, from normal size up to 400%

9 Generating Reports

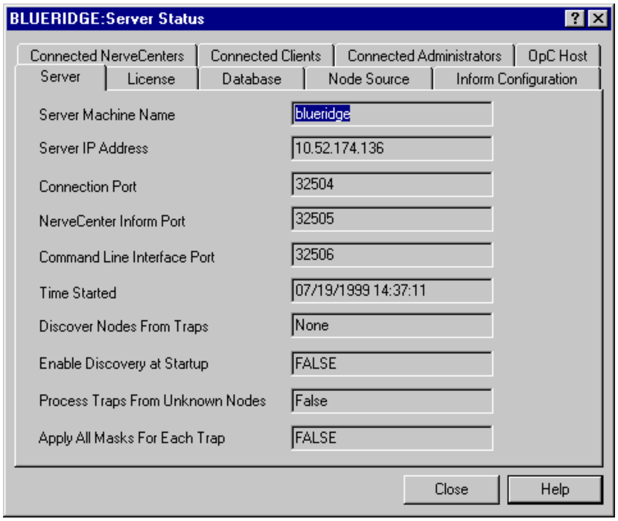
Checking the Status of the Server

Using the NerveCenter Client, you can obtain a good deal of information about the active NerveCenter server. For example, you can check on the status of the machine the server is using as its node source, or the status of the machines running a network management platform that NerveCenter will notify when an Inform alarm action takes place. Or you can display a list of the NerveCenter clients and administrators that are connected to the active server.

TO DISPLAY INFORMATION ABOUT THE ACTIVE SERVER

- Choose **Server Status** from the **Server** menu.

  The Server Status dialog is displayed.



BLUERIDGE:Server Status

Server Machine Name	blueridge
Server IP Address	10.52.174.136
Connection Port	32504
NerveCenter Inform Port	32505
Command Line Interface Port	32506
Time Started	07/19/1999 14:37:11
Discover Nodes From Traps	None
Enable Discovery at Startup	FALSE
Process Traps From Unknown Nodes	False
Apply All Masks For Each Trap	FALSE

The information in this dialog is divided into nine pages, accessible via the nine tabs shown above.

10 Checking the Status of the Server

For an explanation of the information available on a particular page, see the appropriate subsection:

- *Server Tab* on page 160
- *License Tab* on page 162
- *Database Tab* on page 163
- *Node Source Tab* on page 164
- *Inform Configuration Tab* on page 165
- *Connected NerveCenters Tab* on page 166
- *Connected Clients and Connected Administrators Tabs* on page 166
- *OpC Host Tab* on page 167

Server Tab

The Server page presents information about the machine the NerveCenter server is running on, the communication ports on that machine being used by NerveCenter, and the server's node-discovery settings. Table 10-1 provides explanations of the information shown on the Server page:

TABLE 10-1. Fields on Server Tab

Label	Explanation
Server Machine Name	The name of the machine running the active NerveCenter server.
Server IP Address	The IP address of the machine running the active NerveCenter server.
Connection Port	The port the NerveCenter server uses to communicate with the NerveCenter Client.
NerveCenter Inform Port	The port the NerveCenter server uses to receive Inform traps from other NerveCenter servers.
Command Line Interface Port	The port the NerveCenter server uses to communicate with command-line-interface clients.
Time Started	The date and time at which the NerveCenter server was started.

TABLE 10-1. Fields on Server Tab (Continued)

Label	Explanation
Discover Nodes from Traps	Describes how the NerveCenter server is set up to handle traps from nodes that are not in its database. The possible values are All, None, and Filter. ◆ **All**—If the server receives a trap from any node that is not in its database, it adds that node to its database. ◆ **None**—If the server receives a trap from a node that is not in its database, it does not add that node to its database. ◆ **Filter**—If the server receives a trap from a node that is not in its database, it adds that node to its database if the node is allowed by the server's IP address filter. (This filter is shown on the Node Source page.)
No DNS Lookup of Discovered Nodes	False, the default, indicates that NerveCenter will attempt a DNS lookup for any IP address discovered from a trap. True indicates that NerveCenter will not attempt a DNS lookup of any node discovered by a trap. Nodes are added to NerveCenter as an IP address. **Note:** No DNS Lookup of Discovered Nodes is only valid if Discover Nodes from Traps is selected.
Enable Discovery at Startup	True or False. If this attribute is set to True, the NerveCenter server turns the IPSweep alarm on each time you start the server. (The poll TruePoll must be enabled before an instance of the IPSweep alarm can be instantiated.)
Process Traps from Unknown Nodes	True or False. If this attribute is set to True, when the server receives a trap from a node that is not in its database, the server processes that trap—by seeing if it meets the criteria defined in a trap mask—just as it would process a trap from a node in its database.

TABLE 10-2. Fields in the Totaled License Keys Group Box

Label	Explanation
Max Managed Nodes	The maximum number of nodes that all of your licenses permit you to add to the NerveCenter database.
Client Connections	The maximum number of clients that all of your licenses allow you to connect to the NerveCenter server.

License Tab

The License page presents information about your licenses. Table 10-3 and Table 10-2 provides brief explanations of the information shown on the License tab:

TABLE 10-3. Fields in the Selected License Key Group Box

Label	Explanation
License Key	A drop-down list box containing all of your serial numbers that correspond to your licenses.
Start Date	The date on which the selected serial number became effective.
End Date	The date on which the selected serial number expires.
Duration	The time limit for the selected serial number if the product and license were provided as an evaluation copy.
Max Managed Nodes	The maximum number of nodes that the selected license permits you to add to the NerveCenter database.
Currently Managed Nodes	The current number of nodes in the NerveCenter database counted against the selected license.
Client Connections	The maximum number of clients that the selected license allows you to connect to the NerveCenter server.
Serial Number	A serial number for the selected license. You may need this number when dealing with a technical-support representative.
Max Managed Nodes	The maximum number of nodes that all of your licenses permit you to add to the NerveCenter database.
Client Connections	The maximum number of clients that all of your licenses allow you to connect to the NerveCenter server.

Database Tab

The Database page presents information about the NerveCenter database. Table 10-4 provides brief explanations of the information shown on the Database tab:

TABLE 10-4. Fields on Database Tab

Label	Explanation
Database Source Name	The name of your open database connectivity (ODBC) data source.
Machine Name	The name of the host on which the NerveCenter database resides.
Database Name	The full pathname of the NerveCenter database.
Database Status	Indicates whether the server’s connection to the database is currently up or down.
Alarms	The number of alarms in the NerveCenter database.
Polls	The number of polls in the NerveCenter database.
Nodes	The number of nodes in the NerveCenter database.
Masks	The number of trap masks in the NerveCenter database.
Prop. Grps	The number of property groups in the NerveCenter database.
Properties	The number of properties in the NerveCenter database.

Node Source Tab

The Node Source page presents information about the network management platform from which NerveCenter is getting node information and describes NerveCenter's node filters. Table 10-5 provides brief explanations of the information shown on the Node Source tab:

TABLE 10-5. Fields on Node Source Tab

Label	Explanation
Machine	The name of the machine hosting the network management system whose node database is being used by NerveCenter as its source of information about managed nodes.
IP Address	The IP address of the machine hosting the network management system.
Port	The number of the port being used for communication with the platform adapter running on the node-source host.
Connection Status	The current status of the connection between the network management system and the NerveCenter server.
Wanted Capabilities	A list of capabilities—such as isRouter, isHub, isSNMPSupported—that the NerveCenter server uses in filtering the nodes in a network management system's node database. That is, the NerveCenter server will retrieve information only about nodes with one or more of these capabilities.
System Object IDs	A list of OIDs that the NerveCenter server uses in filtering the nodes in a network management system's node database. That is, the NerveCenter server will retrieve information only about nodes whose OIDs match, or are subordinate to, one of the OIDs in this list.
IP Address Filters	A list of IP address ranges that the NerveCenter server uses in filtering the nodes in a network management system's node database. Each entry in the list consists of a subnet address and mask and a list of excluded addresses. For each entry in the list, the NerveCenter server retrieves information about nodes on the subnet that are not specifically excluded.

Inform Configuration Tab

The Inform Configuration page presents information about the network management platforms and NerveCenter servers to which your server is sending Inform messages. Table 10-6 provides brief explanations of the information shown for each Inform destination:

TABLE 10-6. Fields on Inform Configuration Page

Label	Explanation
Status	The current status of the connection between the NerveCenter server and the platform adapter on the machine hosting the network management platform. Or the current status of the connection between the NerveCenter server whose status you're checking and a NerveCenter server to which it is forwarding events.
Machine Name	The name of the machine hosting the network management platform or the destination NerveCenter server.
IP Address	The IP address of that machine.
Port	The port on which the NerveCenter server communicates with the platform adapter or the destination NerveCenter server.
Filter	The filter column summarizes the restrictions on which alarm transitions can cause Inform actions and on what type of information is included in an Inform message: ◆ The first item in this column can be EVENT_ONLY, SYMBOL_ONLY, or EVENT_AND_SYMBOL. This item indicates what type of information will be forwarded to your network management platform: information to be displayed in the event browser, information about map symbol color changes, or both. ◆ The second item is a number representing a severity level. Only transitions whose destination state has a severity level equal to or greater than this value can cause an Inform message to be sent. ◆ Following the severity level, there may be a list of properties. The property group of the node associated with an alarm transition must contain one of these properties before the transition can cause an Inform message to be sent.

Connected NerveCenters Tab

This tab displays a list of the NerveCenter servers that are connected to the NerveCenter server whose status you're checking. These servers can send Inform messages to your server.

Connected Clients and Connected Administrators Tabs

The Connected Clients and Connected Administrator pages list the NerveCenter Clients and NerveCenter Administrators that are currently connected to the NerveCenter server whose status you're checking. Table 10-7 explains what information is presented for each connected Client and Administrator.

TABLE 10-7. Fields on Connected Clients and Connected Administrators Tabs

Label	Explanation
Machine Name	The name of the machine on which the connected Client or Administrator is running.
IP Address	The machine's IP address.
User Name	The name of the user who connected the Client or Administrator to the server. This field may be blank if this person used NerveCenter's unified log-on feature (Windows only).
Time Connected	The date and time at which the user connected to the server.
Access	The group to which the connected user belongs: User or Administrator. Only users with Administrator privileges can write to the NerveCenter database.

OpC Host Tab

If your NerveCenter server is integrated with Hewlett-Packard’s IT/Operations (IT/O), the OpC Host page presents information about the IT/O host and the connection between NerveCenter and IT/O. Table 10-8 provides explanations of the information shown on the OpC Host tab:

TABLE 10-8. Fields on Database Tab

Label	Explanation
Machine	The name of the machine that is hosting IT/O.
IP Address	The machine’s IP address.
Port	The port on which the NerveCenter server and IT/O are communicating.
Connection Status	The current status of the connection between the NerveCenter server and IT/O.

Communications and Data

A

As a tool that comprehensively monitors and manages your network, NerveCenter uses a variety of data transfers to gather, correlate, disseminate, and store information about network events. This appendix outlines the general flow of data into, through, and out of NerveCenter in the course of its operation.

NerveCenter's primary sources of network information are SNMP traps and device responses to NerveCenter polls. If configured appropriately, Open NerveCenter responds to trap and poll data by forwarding it to your network management platform and to other NerveCenters. For example, forwarded event data might ultimately land in a network management platform's Event Categories window or trigger an alarm transition in a central NerveCenter. Although this sequence may happen quickly, the actual communication path from initial receipt of trap or poll data to the final event message has many stages.

As Figure A-1 shows, a trace of the communication path initiated by a managed device's SNMP trap or poll response might look like this:

1. Traps are relayed directly to the NerveCenter Server if the platform and the server are running on different machines. If they're running on the same machine, traps are detected by the operating system trap service or the management platform's trap service and then forwarded to the NerveCenter SNMP Trap process. The NerveCenter SNMP Trap process, in turn, forwards the trap to Open NerveCenter.
2. Open NerveCenter *trap masks* filter incoming traps to see if they are of interest. If a trap is of interest, an internal event, called a *trigger*, is generated and used by active *alarms*. Polls evaluate the poll data returned by managed devices and also use triggers to pass data to alarms.
3. Open NerveCenter alarms correlate the traps and polls with other related data. For example, an alarm might detect that this is the third trap of the same type from the same machine. The alarm then takes any automated actions that were associated with this trap detection. For example, it could issue a trouble ticket or change the device configuration.

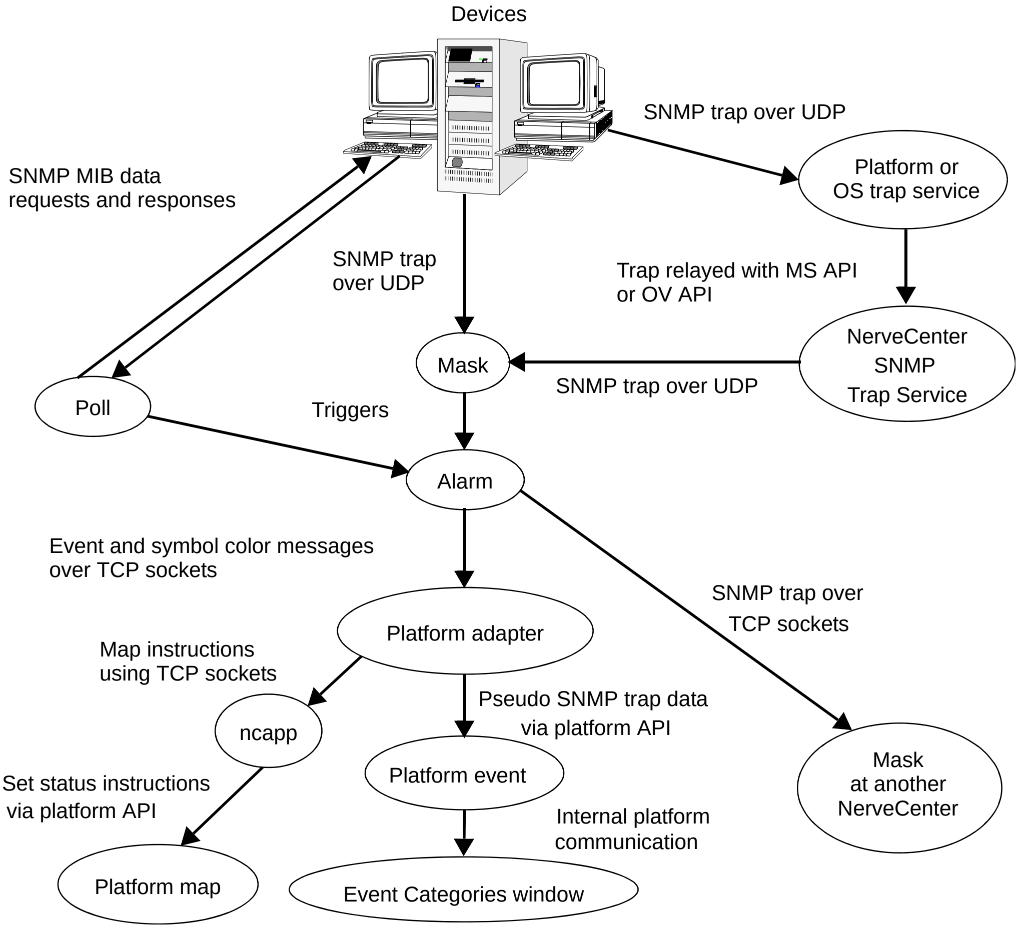

Devices
SNMP trap over UDP
Platform or OS trap service
SNMP MIB data requests and responses
SNMP trap over UDP
Trap relayed with MS API or OV API
NerveCenter SNMP Trap Service
Mask
SNMP trap over UDP
Poll
Triggers
Alarm
Event and symbol color messages over TCP sockets
SNMP trap over TCP sockets
Platform adapter
Map instructions using TCP sockets
ncapp
Pseudo SNMP trap data via platform API
Platform event
Set status instructions via platform API
Internal platform communication
Mask at another NerveCenter
Platform map
Event Categories window

FIGURE A-1. Data Flow

4. If an alarm transition contains the Inform action, the alarm sends a message to the Open NerveCenter platform adapter process, which always resides on the same host as the network management platform, and/or to any listed NerveCenters.

5. The platform adapter determines whether the message requires changing a symbol’s color on the map, initiating an event message, or both. Messages to other NerveCenters forward the trap data.

6. If color changes are required, the platform adapter sends a message to the Open NerveCenter ncapp process, which in turn forwards instructions for color changes to the platform map with an API.

7. If an event is to be posted, the platform adapter uses an API to submit a data structure that resembles an SNMP trap to the platform event facility, which decodes traps, associates text messages with events, and posts them in the Event Categories window.

NerveCenter is a client/server application. The NerveCenter server acts as the hub for the data transfers described in this appendix. As shown in the following illustration, event information moves from managed device to NerveCenter server to management platform. But data also flows between the server and other NerveCenter components in support of this flow.

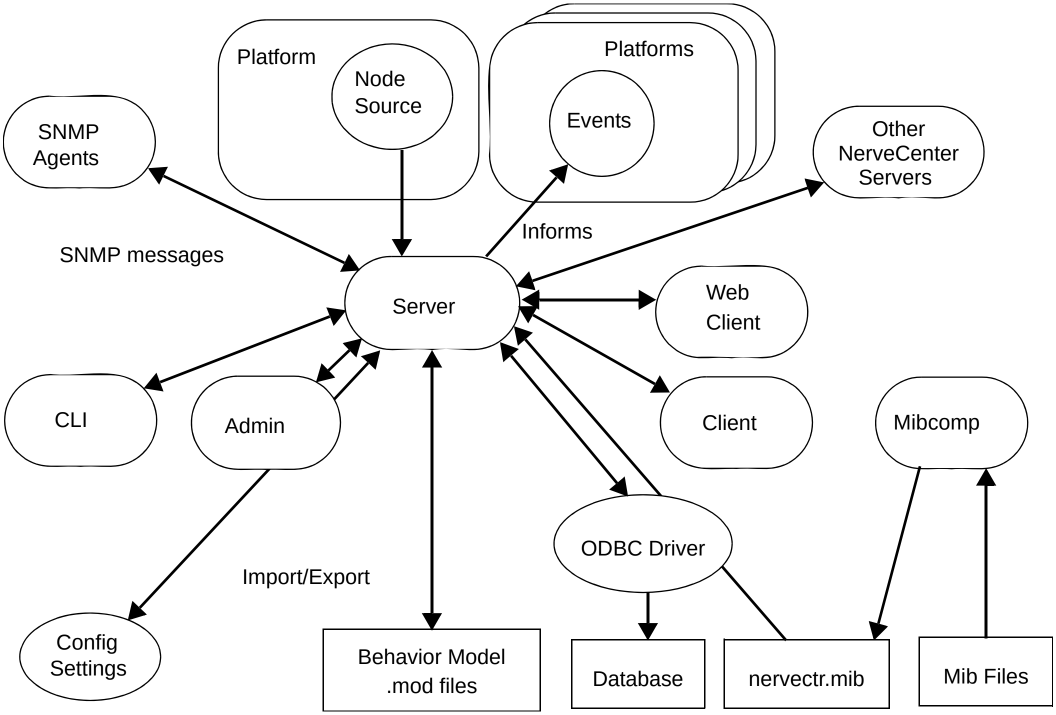


FIGURE A-2. NerveCenter Components

The components shown in the preceding figure are defined in Table A-1:

TABLE A-1. NerveCenter Components

Component	Definition
Client	A user interface to the server. Provides facilities for the creation, modification, maintenance, and monitoring of behavior models.
Web client	A user interface to the server. Meant to be used only for monitoring a network.

TABLE A-1. NerveCenter Components (Continued)

Component	Definition
Administrator	A user interface to the server. Provides facilities for NerveCenter configuration.
Command line interface (CLI)	Provides a subset of client commands for use from the command line, programs, and scripts.
Platform/node source	The network management platform that provides and monitors a list of nodes to be monitored by the server.
Platforms/events	The network management platforms that the server informs as an alarm action.
Other NerveCenters	Other NerveCenter servers that can accept Informs from the server, allowing correlation across multiple domains.
SNMP agents	Agents running on managed nodes that generate traps and respond to NerveCenter polls.
ODBC Driver	The NerveCenter server's interface to its database.
Mibcomp	Utility to compile and merge MIBs into the NerveCenter master MIB.
Configuration Settings	Repository for NerveCenter configuration parameter values—nervecenter.xml configuration file (UNIX) and the Registry (Windows).
Behavior model .mod files	ASCII files containing exported behavior models and their components.

Figure A-3 shows the utilities that install NerveCenter and assist in database management:

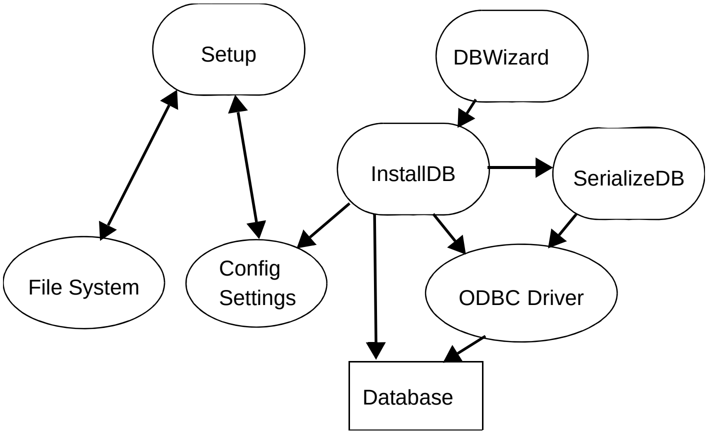


FIGURE A-3. Utilites for Installation and Database Management

The utilities shown in Figure A-3 are defined in Table A-2.

TABLE A-2. NerveCenter Utilities

Utility	Purpose
Setup	Installs the NerveCenter file hierarchy and initializes NerveCenter configuration settings.
DBWizard	GUI for InstallDB.
InstallDB	Command line utility for database creation, initialization, and modification.
SerializeDB	GUI-based utility for importing and exporting database information.
ODBC	The NerveCenter server’s interface to its database.

A Communications and Data

Error Messages

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 B-1. Sections Included in this Appendix

Section	Description
User Interface Messages on page 176	Explains where error messages appear as well as the different types of error messages.
Error Messages on page 178	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:

TABLE B-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 B-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:

- *Action Manager Error Messages* on page 179
- *Alarm Filter Manager Error Messages* on page 183
- *Deserialize Manager Error Messages* on page 184
- *Flatfile Error Messages* on page 184
- *Inform NerveCenter Error Messages* on page 184
- *Inform OV Error Messages* on page 185
- *LogToDatabase Manager Error Messages* on page 187
- *LogToFile Manager Error Messages* on page 188
- *OpC Manager Error Messages* on page 188
- *Poll Manager Error Messages* on page 188
- *Protocol Manager Error Messages* on page 189
- *PA Resync Manager Error Messages* on page 190
- *Server Manager Error Messages* on page 192
- *Trap Manager Error Messages* on page 196
- on page 197
- *OpenView Configuration Error Messages (UNIX)* on page 201

B Error Messages

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 176.

The messages include:

- *Action Manager Error Messages* on page 179
- *Alarm Filter Manager Error Messages* on page 183
- *Deserialize Manager Error Messages* on page 184
- *Flatfile Error Messages* on page 184
- *Inform NerveCenter Error Messages* on page 184
- *Inform OV Error Messages* on page 185
- *LogToDatabase Manager Error Messages* on page 187
- *LogToFile Manager Error Messages* on page 188
- *OpC Manager Error Messages* on page 188
- *Poll Manager Error Messages* on page 188
- *Protocol Manager Error Messages* on page 189
- *PA Resync Manager Error Messages* on page 190
- *Server Manager Error Messages* on page 192
- *Trap Manager Error Messages* on page 196
- *OpenView Configuration Error Messages (UNIX)* on page 201

Action Manager Error Messages

Following is a list of Action Manager error messages.

TABLE B-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.

TABLE B-4. Action Manager Error Messages (Continued)

Error Number	Error	Resolution
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.
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.

TABLE B-4. Action Manager Error Messages (Continued)

Error Number	Error	Resolution
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.
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.

TABLE B-4. Action Manager Error Messages (Continued)

Error Number	Error	Resolution
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.
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.

TABLE B-4. Action Manager Error Messages (Continued)

Error Number	Error	Resolution
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 B-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 B-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 B-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 B-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 B-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.
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.

TABLE B-9. Inform OV Manager Error Messages (Continued)

Error Number	Error	Resolution
1511	*value*. Too many open files.	Close any open files.
1512	*value*. The network subsystem is down.	Reboot the machine.
1513	*value*. The network dropped the connection.	Make sure the peer is running and the network connections are working.
1514	*value*. No buffer space is available.	This might be because you are running several applications, or an application is not releasing resources.
1515	*value*. The network cannot be reached from this host at this time.	Make sure the routers are functioning properly.
1516	*value*. Attempt to connect timed out without establishing a connection.	Make sure the machine is running and on the network.
1517	*value*. The host cannot be found.	Make sure you can ping the host. Check your hosts file or DNS server.
1518	*value*. The network subsystem is unavailable.	Make sure the network services are started on machine.
1519	*value*. Invalid host name specified for destination.	The host name cannot be resolved to an IP address. Enter the name to the hosts file or DNS server.
1520	*value*. The specified address in not available.	Make sure the host name is not zero—try pinging the host.
2001	Command line too long: *value*.	Check the Windows Command Action. Command line exceeds maximum allowed length of 2048 characters.
2006	Fire Trigger Action error: Invalid node name: *value*.	A node name was specified directly in an action and that node doesn't exist in the system.
2007	Fire Trigger Action error: Invalid property name: *value*.	A property was specified directly in an action and that property doesn't exist in the system.
2008	Fire Trigger Action error: Invalid subobject: *value*.	A subobject was specified directly in an action and that subobject doesn't exist in the system.
2009	Inform OV send Packet Failed for platform socket *value*.	

TABLE B-9. Inform OV Manager Error Messages (Continued)

Error Number	Error	Resolution
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 B-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 B-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 B-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 B-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 B-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 B-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.

TABLE B-15. PA Resync Manager Error Messages (Continued)

Error Number	Error	Resolution
1506	%s. The connection was aborted due to timeout or other failure	Make sure the physical network connections are present.
1507	%s. The attempt to connect was refused	Make sure the server is running on the remote host.
1508	%s. The connection was reset by the remote side	Make sure the remote peer is up and running.
1509	%s. A destination address is required	A destination address or host name is required.
1510	%s. The remote host cannot be reached	Make sure the routers are working properly.
1511	%s. Too many open files	Close any open files.
1512	%s. The network subsystem is down	Reboot the machine.
1513	%s. The network dropped the connection	Make sure the peer is running and the network connections are working.
1514	%s. No buffer space is available	This might be because you are running several applications, or an application is not releasing resources.
1515	%s. The network cannot be reached from this host at this time	Make sure the routers are functioning properly.
1516	%s. Attempt to connect timed out without establishing a connection	Make sure the machine is running and on the network.
1517	%s. The host cannot be found	Make sure you can ping the host, check you hosts file or DNS server.
1518	The network subsystem is unavailable	Make sure the network services are started on machine.
1519	%s. Invalid host name specified for destination	The host name cannot be resolved to an IP address. Enter the name to the hosts file or DNS server.
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.

TABLE B-15. PA Resync Manager Error Messages (Continued)

Error Number	Error	Resolution
3500	Connection to %s was successful	N/A

Server Manager Error Messages

Following is a list of Server Manager error messages.

TABLE B-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

TABLE B-16. Server Manager Error Messages (Continued)

Error Number	Error	Resolution
24	Error while reading database. Poll/Mask:*value* uses a simple trigger that doesn't exist in database.	N/A
25	Please report error number *value* to technical support.	N/A
26	User validation failed: Unable to communicate with ncsecurity process :*value.*	~
1001	Windows sockets initialization failed.	Install TCP/IP.
1002	Initialization failed, cannot find ncperl.pl.	Check NCPerl.pl location.
1003	Failed to open MIB: *value.*	Check MIB location.
1004	Failed to parse MIB.	Invalid MIB. Check configuration to see if the correct MIB is specified.
1010	Failed to validate poll: *value*. The poll will be turned off.	Check the poll condition using the Client Application.
1100	*value* (database error).	Try to resolve using the message. If not, call support.
1101	Failed to connect to database. ODBC Connection String in configuration settings is invalid or can't find database server.	Use InstallDB to re-create the ODBC connection string.
1102	Failed to connect to database. ODBC Connection String in configuration settings is empty.	Use InstallDB to re-create the ODBC connection string.
1103	Version table validation failed. NC_Version table doesn't exist in database.	Upgrade the NerveCenter database 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.

TABLE B-16. Server Manager Error Messages (Continued)

Error Number	Error	Resolution
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.
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.

TABLE B-16. Server Manager Error Messages (Continued)

Error Number	Error	Resolution
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.
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.

TABLE B-16. Server Manager Error Messages (Continued)

Error Number	Error	Resolution
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 B-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.	

TABLE B-17. Trap Manager Error Messages (Continued)

Error Number	Error	Resolution
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.
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 B-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.
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.

TABLE B-18. NerveCenter Installation Error Messages (UNIX) (Continued)

Error	Resolution
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 B-18. NerveCenter Installation Error Messages (UNIX) (Continued)

Error	Resolution
Can't create /etc/rc2.d/K94ncservice on Solaris Can't create /etc/rc2.d/K940ncservice on HP-UX	The script couldn't create the Open NerveCenter rc script /etc/rc2.d/K94ncservice on Solaris or 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 B-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.

B Error Messages

Index

I

Index