



LogMatrix
NerveCenter

Behavior Models Cookbook

**UNIX and Windows
Version 5.1.0***

Copyright

Portions ©1989-2011 LogMatrix, Inc. All rights reserved.

Disclaimers

LogMatrix, Inc. (“LogMatrix”) makes no representations or warranties, either expressed or implied, by or with respect to anything in this manual, and shall not be liable for any implied warranties of merchantability or fitness for a particular purpose or for any indirect, special or consequential damages.

These applications are available through separate, individual licenses. Not every feature or application described herein is licensed to every customer. Please contact LogMatrix if you have licensing questions.

No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, photocopying, recording or otherwise, without prior written consent of LogMatrix. While every precaution has been taken in the preparation of this book, LogMatrix assumes no responsibility for errors or omissions. This publication and the features described herein are subject to change without notice.

The program and information contained herein are licensed only pursuant to a license agreement that contains use, reverse engineering, disclosure and other restrictions.

Trademarks

LogMatrix is registered in the U.S. Patent and Trademark Office. NerveCenter and the LogMatrix Logo are trademarks of LogMatrix, Inc.

All other products or services mentioned in this manual may be covered by the trademarks, service marks, or product names as designated by the companies who market those products.

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

1 Introduction

Overview of this book	2
NerveCenter Documentation	3
Using the Online Help	3
Printing the Documentation	3
The NerveCenter Documentation Library	4
UNIX Systems	5
Document Conventions	5
Documentation Feedback	6
LogMatrix Technical Support	6
Professional Services	6
Educational Services	7
Contacting the Customer Support Center	7
For Telephone Support	7
For E-mail Support	7
For Electronic Support	7
For Online KnowledgeBase Access	7
For User Community Access	8

2 About Behavior Models

Using Behavior Models	9
Using Vendor-Specific Behavior Models	10
Customizing NerveCenter's predefined behavior models	11

3 Authentication Models

Authentication	15
--------------------------	----

4 Data Collection Models

AllTraps_LogToFile	18
GatewayList	19

	IfData_LogToFile	20
	IPSweep (UNIX)	21
	IPSweep (Windows)	23
	NoSystemInfo	25
	RoutingMon	26
5	DBI Models	
	ifEntry_LogToDBI	27
6	Host Resource Models	
	DiskUsage	29
	MemUsage	30
	ProcessorLoad	31
7	Interface Status Models	
	IfErrorStatus	34
	IfLinkUpDown	36
	IfLoad	38
	IfUpDownStatus	40
	IfUpDownStatusByType	42
8	Node Status Models	
	NodeStatus	47
	NodeStatusDwnStrm	49
9	TCP Status Models	
	TcpConnMon	51
	TcpRetransAlg	53
	TcpRetransMon	55
10	Troubleshooting Models	
	InformConnectStatus	57
11	Vendor Models	
	Cisco (Catalyst Switch Models)	62

CatalystAlarm	62
CatalystFanStatus	63
CatalystModuleStatus	64
CatalystPs1Status, CatalystPs2Status, CatalystPsTestResult	65
Cisco (Environmental Monitoring Models)	66
CiscoConfigChange	66
CiscoCpuUtilization	67
CiscoFanStatus	68
CiscoMemoryPoolStatus	69
CiscoPwrSupplyStatus	70
CiscoTempStatus	71
CiscoVoltageStatus	72
Cisco (Legacy Models)	73
CiscoBufferUtilization	73
CiscoCPUUtilization	75
CiscoCRCCheck	77
CiscoFanCheck	79
CiscoMemGetCheck	80
CiscoPacketsDrop	82
CiscoPwrSupplyCheck	84
CiscoTempCheck	86
Compaq Models	88
CpqDaLogDrvStat	88
CpqNic	90
CpqDaPhyDrvThresh	91
CpqScsiPhyStat	92
CpqUnknown	94
CpqUps	94
SNMP Research Models	97
SnmresCritAppStatus	97
SnmresDiskUsage	99
SnmresGenericLogMonitor	101
SnmresLoginMonitor	103
SnmresMemUsage	105
SnmresProcessorLoad	106
Wellfleet (Nortel) Models	108
WfBufferUtilization	108
WfFanCheck	110
WfMemCheck	111
WfPwrSupplyCheck	113
WfTempCheck	115



Index..... 117

Welcome to *Behavior Models Cookbook*. 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
<i>Overview of this book on page 2</i>	Includes an overview of the contents of this guide and what you need to know before you use the guide.
<i>NerveCenter Documentation on page 3</i>	Lists and describes the components of the LogMatrix NerveCenter support system, including printed guides, online guides, help, and links to the LogMatrix NerveCenter Web site and the LogMatrix technical support Web site.
<i>LogMatrix Technical Support on page 6</i>	Describes how to access the NerveCenter knowledge base and other LogMatrix support services.

Overview of this book

The *Behavior Models Cookbook* describes behavior models shipped with LogMatrix NerveCenter. Behavior models detect network conditions or events, correlate related conditions, and invoke appropriate actions in response. NerveCenter comes configured for immediate use, but you can easily customize it to meet the demands of your network environment.

Before reading this manual, you should be familiar with the Simple Network Management Protocol (SNMP), common MIB-II objects, your network management platform, and Perl scripting.

This book contains the following chapters:

Title	Description
<i>Chapter 2, About Behavior Models</i>	Describes how to use behavior models shipped with LogMatrix NerveCenter.
<i>Chapter 3, Authentication Models</i>	Discusses the network authentication behavior model.
<i>Chapter 4, Data Collection Models</i>	Describes various behavior models that perform information gathering tasks: list gateways, list devices missing system information, log information from SNMP traps, identify routing errors, log interface attribute values for ifEntry nodes, identify nodes using a ping sweep.
<i>Chapter 5, DBI Models</i>	Describes the behavior model that uses Perl to query and write to a database.
<i>Chapter 7, Interface Status Models</i>	Describes behavior models that monitors interface error rates, traffic, and status.
<i>Chapter 8, Node Status Models</i>	Describes the behavior models used for monitoring node status and for downstream alarm suppression.
<i>Chapter 9, TCP Status Models</i>	Discusses the behavior models used to monitor a network's TCP connections and transmissions.
<i>Chapter 10, Troubleshooting Models</i>	Discusses the behavior model used to monitor the status of NerveCenter's connection with the OpenView Platform Adapter (OVPA) and the Inform Acknowledgement Queue.
<i>Chapter 11, Vendor Models</i>	Describes the groups of behavior models for use in monitoring Cisco and Wellfleet (Nortel) routers, and Compaq devices.

NerveCenter Documentation

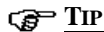
This section describes the available NerveCenter documentation, which explains important concepts in depth, describes how to use NerveCenter, and provides answers to specific questions.


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

**TIP**

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

Printing the Documentation

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

**NOTE**

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

The NerveCenter Documentation Library

The following documents ship with NerveCenter.

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

UNIX Systems

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

Document Conventions

This document uses the following typographical conventions:

Element	Convention	Example
Key names, button names, menu names, command names, and user entries	Bold	Press Tab Enter ovpa -pc
<ul style="list-style-type: none"> ◆ A variable you substitute with a specific entry ◆ Emphasis ◆ Heading or Publication Title 	<i>Italic</i>	Enter <i>./installdb -f IDBfile</i>
Code samples, code to enter, or application output	Code	<code>iifInOctets > 0</code>
Messages in application dialog boxes	Message	Are you sure you want to delete?
An arrow (>) indicates a menu selection	>	Choose Start > Programs > OpenService NerveCenter
A link to a section in the same book	<i>Blue Italic</i>	For more information, see <i>Correlating Conditions</i> .
A link to a section in a different book	<i>Green Italic</i>	For more information, see <i>Correlating Conditions</i> in <i>Monitoring Your Network with NerveCenter</i> .



CAUTION

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

**NOTE**

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

**TIP**

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



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

Documentation Feedback

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

Documentation
LogMatrix, Inc.
6'O qwpvTq{ criC xg. "Uwkg"472
Marlborough, MA 01752

documentation@logmatrix.com

LogMatrix Technical Support

LogMatrix is committed to offering the industry's best technical support to our customers and partners. You can quickly and easily obtain support for NerveCenter, our proactive IT management software0

Professional Services

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

Educational Services

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

Contacting the Customer Support Center

For Telephone Support

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

For E-mail Support

E-mail: techsupport@logmatrix.com.

For Electronic Support

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

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

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

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

For Online KnowledgeBase Access

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

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

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

For User Community Access

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

This chapter describes how to use the behavior model shipped with NerveCenter.



NOTE

These behavior models are for demonstration and example purposes only, and not intended for production use.

Using Behavior Models

If you need a vendor-specific behavior model (Cisco, Compaq, or Wellfleet (Nortel)), go to the section *Using Vendor-Specific Behavior Models on page 10*. Otherwise, perform these steps to use this behavior model.

TO USE A PREDEFINED BEHAVIOR MODEL

1. Import the model. (See *Importing Node, Object, and Behavior Model Files in Designing and Managing Behavior Models* for details.)
 2. (Optional) Choose to customize certain settings for the model, such as poll and timer intervals, counters, thresholds, alarm actions, and so on. See the section *Optional customization* in the description of each behavior model for suggestions on customizing a behavior model. For details about how to customize various elements of the behavior model, see *Designing and Managing Behavior Models*.
 3. Turn on the model.
 - a. From the NerveCenter Client's **Admin** menu, choose **Alarm Definition List**.
 - b. Right-click the model's alarm(s) and choose **On**.
-

Using Vendor-Specific Behavior Models

To use this vendor-specific behavior model (Cisco, Compaq, or Wellfleet (Nortel)), perform these steps:

TO USE A VENDOR-SPECIFIC BEHAVIOR MODEL

1. Decide which vendor's base objects you want to monitor and note each object's .asn file under

UNIX	Windows
mibs/vendors/ <i>vendor-name</i>	Mib\vendors\ <i>vendor-name</i>

2. Add or remove the necessary MIB definitions in:

UNIX	Windows
mibs/mibcomp.txt	Mib\mibcomp.txt

3. Compile the NerveCenter MIB.
4. Reload the NerveCenter MIB.

See *Managing Management Information Bases (MIBs) in Managing NerveCenter* for details about working with MIBs.

Follow the steps in the preceding section, *Using Behavior Models on page 9*.

Customizing NerveCenter's predefined behavior models

NerveCenter behavior models contain all the required mask, poll, alarm, and property group definitions for basic network management using MIB-II objects. With a little customization, you can collect data such as node status, security breaches, performance statistics, and fault, load, and error conditions.

By default, NerveCenter stores its behavior models as .mod files in the NerveCenter installation/Model directory. *Table 2-1* lists all the models which ship with NerveCenter.

TABLE 2-1. NerveCenter Pre-Defined Behavior Models

Model Type	Model Description Location
Authentication	<i>Authentication on page 15</i>
Data Collection	<i>AllTraps_LogToFile on page 18</i> <i>GatewayList on page 19</i> <i>IfData_LogToFile on page 20</i> <i>IPSweep (UNIX) on page 21</i> <i>IPSweep (Windows) on page 23</i> <i>NoSystemInfo on page 25</i> <i>RoutingMon on page 26</i>
DBI	<i>ifEntry_LogToDBI on page 27</i>
Interface Status	<i>IfErrorStatus on page 34</i> <i>IfLinkUpDown on page 36</i> <i>IfLoad on page 38</i> <i>IfUpDownStatus on page 40</i> <i>IfUpDownStatusByType on page 42</i>
Node Status	<i>NodeStatus on page 47</i> <i>NodeStatusDwnStrm on page 49</i>
TCP Status	<i>TcpConnMon on page 51</i> <i>TcpRetransAlg on page 53</i> <i>TcpRetransMon on page 55</i>
Troubleshooting	<i>InformConnectStatus on page 57</i>

TABLE 2-1. NerveCenter Pre-Defined Behavior Models

Model Type	Model Description Location
Vendor - Cisco	<i>CiscoBufferUtilization on page 73</i>
	<i>CiscoCPUUtilization on page 75</i>
	<i>CiscoCRCCheck on page 77</i>
	<i>CiscoFanCheck on page 79</i>
	<i>CiscoMemGetCheck on page 80</i>
	<i>CiscoPacketsDrop on page 82</i>
	<i>CiscoPwrSupplyCheck on page 84</i>
	<i>CiscoTempCheck on page 86</i>
Vendor - Compaq (HP)	<i>CpqDaLogDrvStat on page 88</i>
	<i>CpqNic on page 90</i>
	<i>CpqDaPhyDrvThresh on page 91</i>
	<i>CpqScsiPhyStat on page 92</i>
	<i>CpqUnknown on page 94</i>
	<i>CpqUps on page 94</i>
Vendor - SNMP Research v3	<i>SnmpresCritAppStatus on page 97</i>
	<i>SnmpresDiskUsage on page 99</i>
	<i>SnmpresGenericLogMonitor on page 101</i>
	<i>SnmpresLoginMonitor on page 103</i>
	<i>SnmpresMemUsage on page 105</i>
	<i>SnmpresProcessorLoad on page 106</i>
Vendor - Wellfleet (Nortel Networks)	<i>WfBufferUtilization on page 108</i>
	<i>WfFanCheck on page 110</i>
	<i>WfMemCheck on page 111</i>
	<i>WfPwrSupplyCheck on page 113</i>
	<i>WfTempCheck on page 115</i>

For more information about NerveCenter's behavior models, refer to *Designing and Managing Behavior Models*. You can also view the associated Notes for alarms, polls, and masks.

You may want to modify the behavior models in the following ways:

- ◆ Change a poll or alarm to target a subset of devices.
For details, see *Defining Property Groups and Properties in Designing and Managing Behavior Models*.
- ◆ Provide a new polling interval.
For details, see *Defining a Poll in Designing and Managing Behavior Models*.
- ◆ Alter the network conditions monitored by a poll.
For details, see *Writing a Poll Condition in Designing and Managing Behavior Models*.
- ◆ Modify an alarm definition to respond differently to detected network conditions.
For details, see *Using Alarms in Designing and Managing Behavior Models*.
- ◆ Modify an Inform event message.
For details, see *Inform* or *Inform Specific Numbers in Designing and Managing Behavior Models*.

The authentication models reside under models/authentication, and consist of:

Section	Description
Authentication on page 15	Tracks authentication failures on the network.

Authentication

Path: /model/authentication/authentication.mod

This behavior model tracks authentication failures on the network. If more than three failures occur within ten minutes, then NerveCenter sends a 1101 Inform to the network management platform.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ NO_PROP

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ AuthFail (poll)
- ◆ AuthQuickFail (poll)
- ◆ AuthFail (mask)
- ◆ Authentication (alarm)


Events Sent to OpenView: [Table 3-1](#) summarizes the events this behavior model sends to OpenView:

TABLE 3-1. Events Sent to OpenView by Authentication

Event Name	Inform Number	Description
NC_Authentication	1101	This inform event is triggered by the Authentication alarm and indicates that three consecutive log in attempts failed for the node. This event could indicate that unauthorized access to the network has been attempted.

For more information about integrating NerveCenter with OpenView or the other NerveCenter-supported network manager platforms, refer to *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. Whenever you import a model, NerveCenter will overwrite objects with the same name.

- ◆ Set timers for persistence

See *Defining a Poll in Designing and Managing Behavior Models* about setting the poll rate or *Fire Trigger in Designing and Managing Behavior Models* about setting a delay time for details.

- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIbus

See *Alarm Actions in Designing and Managing Behavior Models* for details.

The data collection models reside under models/data_collection and consist of:

Section	Description
<i>AllTraps_LogToFile on page 18</i>	Repeatedly logs the information contained in all SNMP traps detected by NerveCenter.
<i>GatewayList on page 19</i>	Identifies gateways and outputs them in the NerveCenter Client's Alarm Summary window.
<i>IfData_LogToFile on page 20</i>	Repeatedly logs noteworthy interface attribute values for every managed, non-suppressed node that contains the ifEntry property.
<i>IPSweep (UNIX) on page 21</i>	(UNIX version) Identifies responsive nodes in the network that are not in the Open NerveCenter database.
<i>IPSweep (Windows) on page 23</i>	(Windows version) Identifies responsive nodes in the network that are not in the Open NerveCenter database.
<i>NoSystemInfo on page 25</i>	Determines which devices are missing system information (name, description, and contact).
<i>RoutingMon on page 26</i>	Monitors routing activity by tracking nodes that cross specified thresholds for: <ul style="list-style-type: none">◆ ICMP Destination Unreachable messages◆ Time exceeded messages◆ IP datagrams discarded because no route could be found to deliver them

AllTraps_LogToFile

Path: /model/data_collection/alltraps_logtofile.mod

This Behavior model repeatedly logs the information contained in all SNMP traps detected by NerveCenter. Each alarm instance represents a single node.

On each transition, NerveCenter performs a “Log To File” alarm action so that reports on SNMP traps can be produced.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ NO_PROP

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ AllTraps (mask)

Events Sent to OpenView: AllTraps_LogToFile sends no events to OpenView.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

TIP

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Log To File alarm action options (file pathname, log action variables, verbose or non-verbose output).

See *Log to File in Designing and Managing Behavior Models* for details.

GatewayList

Path: /model/data_collection/gatewaylist.mod

This behavior model identifies gateways—devices that forward IP packets—in a managed network. NerveCenter outputs all identified gateways in the NerveCenter Client’s Alarm Summary window.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ NO_PROP

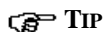
Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ IfGateway (poll)

Events Sent to OpenView: GatewayList sends no events to OpenView.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.



TIP

It’s a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Poll intervals
See *Defining a Poll in Designing and Managing Behavior Models* for details.
- ◆ Set poll conditions to detect additional information, such as gateways with high traffic, gateways with errors, and so on.
See *Writing a Poll Condition in Designing and Managing Behavior Models* for details.
- ◆ Add a NerveCenter alarm action to write the list of gateways to a log or to an xterm window.
See *Alarm Actions in Designing and Managing Behavior Models* for details.

IfData_LogToFile

Path: /model/data_collection/ifdata_logtofile.mod

This Behavior model repeatedly logs noteworthy interface attribute values for every managed, non-suppressed node that contains the ifEntry property.

On each transition, NerveCenter performs a “Log To File” alarm action so that reports on interface attribute values can be produced.

Property:

The following property must be in the property group for the nodes you want to monitor:

- ◆ NO_PROP


Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ IfData (poll)

Events Sent to OpenView: IfData_LogToFile sends no events to OpenView.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Change ifEntry attributes polled to suit your own network management strategy.
See *Writing a Poll Condition in Designing and Managing Behavior Models* for details.
- ◆ Log To File alarm action options (file pathname, log action variables, verbose or non-verbose output)
See *Log to File in Designing and Managing Behavior Models* for details.

IPSweep (UNIX)

Path: /model/data_collection/ipsweep_unix.mod

(UNIX version) This behavior model identifies responsive nodes in the network that are not in the Open NerveCenter database. When NerveCenter finds such a node, it sends an SNMP trap to either the node source (if one is specified in NerveCenter) or the local host, so that ipsweep can either prompt population of the node database in the network management platform or in Open NerveCenter.

IPSweep runs the ipsweep executable whenever the alarm transitions to the PingSweep state. By default, the path for ipsweep is /opt/OSInc/bin.

If IPSweep does not reside on this default path, you will have to modify the two Command alarm action instances in the alarm definition for IPSweep with the correct path. For information on the Command alarm action, refer to *Command in Designing and Managing Behavior Models*.



NOTE

If you are using the IPSweep alarm to populate a network management platform's database, turn off your network management platform's network topology discovery. See *Using IPSweep Behavior Model in Designing and Managing Behavior Models* for details.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ NO_PROP

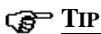
Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ TruePoll (poll)
- ◆ IPSweep (alarm)

Events Sent to OpenView: IPSweep sends no events to OpenView.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.



TIP

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Set the timer to perform the ping sweep

See *Fire Trigger in Designing and Managing Behavior Models* for details about changing the delay time, or *Using IPSweep Behavior Model in Designing and Managing Behavior Models*.

- ◆ Log To File alarm action options (file pathname, log action variables, verbose or non-verbose output)

See *Log to File in Designing and Managing Behavior Models* for details.

IPSweep (Windows)

Path: \model\data_collection\ipsweep_nt.mod

(Windows version) This behavior model identifies responsive nodes in the network that are not in the Open NerveCenter database. When NerveCenter finds such a node, it sends an SNMP trap to either the node source (if one is specified in NerveCenter) or to the local host, so that IPSweep.EXE can either prompt population of the node database in the network management platform or in Open NerveCenter.

IPSweep runs IPSweep.exe whenever the alarm transitions to the PingSweep state. By default, the path for IPSweep.exe is C:\program files\OpenService\NerveCenter\bin.

If IPSweep.exe does not reside on this default path, you will have to modify the two Command alarm action instances in the alarm definition for IPSweep with the correct path. For information on the Command alarm action, refer to *Command in Designing and Managing Behavior Models*.



NOTE

If you are using the IPSweep alarm to populate a network management platform's database, turn off your network management platform's network topology discovery. See *Using IPSweep Behavior Model in Designing and Managing Behavior Models* for details.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ NO_PROP

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ TruePoll (poll)
- ◆ IPSweep (alarm)

Events Sent to OpenView: IPSweep sends no events to OpenView.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.



TIP

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Set the timer to perform the ping sweep

See *Fire Trigger in Designing and Managing Behavior Models* for details about changing the delay time, or *Using IPSweep Behavior Model in Designing and Managing Behavior Models*.

- ◆ Log To File alarm action options (file pathname, log action variables, verbose or non-verbose output)

See *Log to File in Designing and Managing Behavior Models* for details.

NoSystemInfo

Path: /model/data_collection/nosysteminfo.mod

This behavior model determines which devices are missing system information (name, description, and contact). If you assign trouble tickets based on the value in the contact field, for example, that information is very important.

You can use this model as a template to create other alarms that verify the presence of particular MIB data that you require in your environment.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ NO_PROP

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ GotSystemInfo (poll)

Events Sent to OpenView: NoSystemInfo sends no events to OpenView.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

TIP

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Poll intervals
See *Defining a Poll in Designing and Managing Behavior Models* for details.
- ◆ If a subset of managed nodes is required to have system data initialized, change the GotSystemInfo poll's property to target those devices.
See *Defining a Poll in Designing and Managing Behavior Models* for details.
- ◆ Add actions to the noSystemInfo transition to log the occurrence or send e-mail to the person responsible for maintaining the agent's system data. Add an SNMP Set action to set the system data if the attribute values are known and the agents are configured to allow writes.
See *Associating an Action with a Transition in Designing and Managing Behavior Models* for details.

RoutingMon

Path: /model/data_collection/routingmon.mod

This behavior model monitors routing activity by tracking nodes that cross specified thresholds for:

- ♦ ICMP Destination Unreachable messages
- ♦ Time exceeded messages
- ♦ IP datagrams discarded because no route could be found to deliver them

Property: The following property must be in the property group for the nodes you want to monitor:

- ♦ NO_PROP


Trigger Generators: Below is a list of the objects that generate triggers:

- ♦ RoutingDestUnreach (poll)
- ♦ RoutingNoRoutes (poll)
- ♦ RoutingTTLExpired (poll)

Events Sent to OpenView: RoutingMon sends no events to OpenView.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ♦ Poll intervals
See *Defining a Poll in Designing and Managing Behavior Models* for details.
- ♦ Poll conditions for ICMP message and IP datagram thresholds
See *Writing a Poll Condition in Designing and Managing Behavior Models* for details.

The DBI models reside under models/DBI, and consist of:

Section	Description
<i>ifEntry_LogToDBI on page 27</i>	Collects ifEntry metrics and logs them to a database.

ifEntry_LogToDBI

Path: /model/DBI/ifEntry_LogToDBI.mod

Collects ifEntry metrics and logs them to a database.

The model takes advantage of NerveCenter's single threaded Perl interpreter by establishing a connection and allowing each successive use (whether the same node or not) to utilize the connection. Only in the event when a connection is dropped is an attempt made to re-establish.

The ifEntry data is collected via normal polls and then inserted into the database.

The table schema is as follows:

TABLE 5-1. Schema for IfEntry_LogToDBI

Field	Type	Null	Key	Default	Extra
ipAddress	varchar(16)	YES		NULL	
ifIndex	int(11)	YES		NULL	
ifInOctets	int(11)	YES		NULL	
ifOutOctets	int(11)	YES		NULL	
ifSpeed	int(11)	YES		NULL	
timestamp	int(11)	YES		NULL	

**NOTE**

The ifEntry_LogToDBI alarm works only with mySQL database.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ ifEntry

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ DBI_OK (poll)
- ◆ DBI_NoConnect (poll)

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

**TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Edit the ifEntry_LogToDBI Poll to modify the following:
 - ◆ Update username and password information
 - ◆ Add/Delete ifEntry metrics (and update database table)
 - ◆ Update DSN script to utilize a different database
- ◆ Poll intervals

See *Defining a Poll in Designing and Managing Behavior Models* for details.

The Host Resource models reside under `models/host_resource`, and consist of:

Section	Description
<i>DiskUsage</i>	Monitors disk usage on a server that supports the host resource mib.
<i>MemUsage</i>	Monitors memory usage on a server
<i>ProcessorLoad</i>	Monitors the processor load on a server

DiskUsage

Path: `/model/host_resource/diskusage.mod`

This behavior model monitors disk usage on a server that supports the host resource mib. It detects sustained conditions (15 minutes) of High and Very High disk usage for each disk in the server.

Currently there are no actions in the transition for logging or informs.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ `hrStorageEntry`

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ `DiskUsagePoll` (poll)
- ◆ `DiskUsageFastPoll` (poll)

Events Sent to OpenView: Currently there are no events sent to OpenView from this model.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

TIP

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Add log to file to the transitions to meet your specific site requirements
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.
- ◆ Add the appropriate platform inform actions to the DiskUsageNotify transitions
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.
- ◆ Adjust the thresholds in the trigger generators for the disk high and very high conditions
See *Writing a Poll Condition in Designing and Managing Behavior Models* for details.
- ◆ Adjust the timing on the transitions firetrigger used to verify the threshold violation over a sustained period of time
See *Fire Trigger in Designing and Managing Behavior Models* and *Writing a Poll Condition in Designing and Managing Behavior Models* for details.

MemUsage

Path: /model/host_resource/memusage.mod

This behavior model monitors memory usage on a server that supports the host resource mib. It detects sustained conditions (15 minutes) of High and Very High memory usage for the server.

Currently there are no actions in the transition for logging or informs.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ hrStorageEntry


Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ MemUsagePoll (poll)

Events Sent to OpenView: Currently there are no events sent to OpenView from this model.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Add log to file to the transitions to meet your specific site requirements
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.
- ◆ Add the appropriate platform inform actions to the MemUsageNotify transitions
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.
- ◆ Adjust the thresholds in the trigger generators for the memory high and very high conditions
See *Writing a Poll Condition in Designing and Managing Behavior Models* for details.
- ◆ Adjust the timing on the transitions firetrigger used to verify the threshold violation over a sustained period of time
See *Fire Trigger in Designing and Managing Behavior Models* and *Writing a Poll Condition in Designing and Managing Behavior Models* for details.

ProcessorLoad

Path: \model\host_resource\processorload.mod

This behavior model monitors cpu usage on a server that supports the host resource mib. It detects sustained conditions (15 minutes) of High and Very High cpu usage for the server.

Currently there are no actions in the transition for logging or informs.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ hrStorageEntry

Trigger Generators: Below is a list of the objects that generate triggers:


- ◆ ProcessorLoadPoll (poll)
- ◆ ProcessorLoadFastPoll (poll)

Events Sent to OpenView: Currently there are no events sent to OpenView from this model.

For complete information about events sent to OpenView, see the `trapd.conf.txt` file shipped with these behavior models residing in `\model\tcp_status`.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Add log to file to the transitions to meet your specific site requirements

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.

- ◆ Add the appropriate platform inform actions to the CPUHiNotify transitions

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

- ◆ Adjust the thresholds in the trigger generators for the cpu high and very high conditions

See *Writing a Poll Condition in Designing and Managing Behavior Models* for details.

- ◆ Adjust the timing on the transitions firetrigger used to verify the threshold violation over a sustained period of time

See *Fire Trigger in Designing and Managing Behavior Models* and *Writing a Poll Condition in Designing and Managing Behavior Models* for details.

The interface status models reside under `models/interface_status`, and consist of:

Section	Description
<i>IfErrorStatus on page 34</i>	Monitors error rates on interfaces.
<i>IfLinkUpDown on page 36</i>	Monitors node interfaces.
<i>IfLoad on page 38</i>	Determines whether device load is low, medium, or high.
<i>IfUpDownStatus on page 40</i>	Determines whether each interface on a managed node is up, down, or testing.
<i>IfUpDownStatusByType on page 42</i>	Manages the following types of interfaces: <ul style="list-style-type: none">◆ Asynchronous Transfer Mode (ATM)◆ Integrated Services Digital Network (ISDN)◆ Fiber Distributed Data Interface (FDDI)◆ Frame Relay Permanent Virtual Circuit (PVC) subinterfaces◆ Frame Relay◆ Local Area Network (LAN)◆ Switched Multimegabit Data Service (SMDS)◆ Synchronous Optical Network (SONET)◆ Wide Area Network (WAN)

IfErrorStatus

Path: /model/interface_status/iferrorstatus.mod

This multi-alarm behavior model monitors error rates on interfaces. When the interface error rates are high, NerveCenter sends a 7000 Inform to the platform. If a high error rate persists, NerveCenter sends a 7001 Inform to the platform. You can trigger other alarms that use vendor-specific MIBs to further refine the problem and its cause.

Two additional alarms, IfErrorStatus_LogToDB and IfErrorStatus_LogToFile, are part of this model. When you enable these alarms, they perform a Log To Database or a Log to File alarm action, respectively, on each transition so that reports on interface error rates can be produced.



NOTE

The LogToDB alarm works on Windows only.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ ifEntry

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ IfErrorRates (poll)


Events Sent to OpenView: [Table 7-1](#) summarizes the events this behavior model sends to OpenView:

TABLE 7-1. Events Sent to OpenView by IfErrorStatus

Event Name	Inform Number	Description
NC_Interface_Error	7000	The error rate for this interface is high.
NC_Interface_ErrPers	7001	The error rate for this interface has been high over a persistent period of time.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Set counters for persistence

See *Alarm Counter in Designing and Managing Behavior Models* for details. See *Defining a Poll in Designing and Managing Behavior Models* about setting the poll rate or *Fire Trigger in Designing and Managing Behavior Models* about setting a delay time for details.

- ◆ Poll intervals

See *Defining a Poll in Designing and Managing Behavior Models* for details.

- ◆ Poll conditions for error rates

See *Writing a Poll Condition in Designing and Managing Behavior Models* for details.

- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIBus

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

- ◆ Log To File alarm action options
(file pathname, log action variables, verbose or non-verbose output)

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.

- ◆ Log To Database alarm action options
(log identity number and log action variables)

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to Database in Designing and Managing Behavior Models* for details.

IfLinkUpDown

Path: /model/interface_status/iflinkupdown.mod

This multi-alarm behavior model monitors node interfaces. When an interface goes down and remains down for a specified time (the default is three minutes), NerveCenter sends a 7004 Inform to the platform. When an interface comes back up, the alarms return to ground. The monitored nodes must be associated with a property group that contains the property ifEntry.

Two additional alarms, IfLinkUpDown_LogToDB and IfLinkUpDown_LogToFile, are part of this model. When you enable these alarms, they perform a Log To Database or a Log to File alarm action, respectively, on each transition so that reports on interface up/down status can be produced.



NOTE

The LogToDB alarm works on Windows only.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ ifEntry

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ LinkDown (mask)
- ◆ LinkUp (mask)
- ◆ IfLinkUpDown (alarm)
- ◆ IfLinkUpDown_LogToDB (alarm)
- ◆ IfLinkUpDown_LogToFile (alarm)


Events Sent to OpenView: [Table 7-2](#) summarizes the events this behavior model sends to OpenView:

TABLE 7-2. Events Sent to OpenView by IfLinkUpDown

Event Name	Inform Number	Description
NC_Link_Down	7004	Linkdown trap received from interface.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Set timers for persistence

See *Defining a Poll in Designing and Managing Behavior Models* about setting the poll rate or *Fire Trigger in Designing and Managing Behavior Models* about setting a delay time for details.

- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIbus

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

- ◆ Log To File alarm action options
(file pathname, log action variables, verbose or non-verbose output)

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.

- ◆ Log To Database alarm action options (log identity number and log action variables)

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to Database in Designing and Managing Behavior Models* for details.

IfLoad

Path: /model/interface_status/ifload.mod

Determines whether device load is low, medium, or high. Load is the amount of interface traffic compared to the media's capacity, expressed as a percentage. There are two definitions for each—one for point-to-point networks and one for broadcast networks.

This model can give an immediate impression of network utilization; however, it can also be used to track trends based on time, day, or location; to identify runaway processes that are using network resources; to help plan for network reconfiguration; and to help in any task that requires knowledge of traffic levels.

Two additional alarms, IfLoad_LogToDB and IfLoad_LogToFile, are part of this model. When you enable these alarms, they perform a Log To Database or a Log to File alarm action, respectively, on each transition so that reports on device load levels can be produced.



NOTE

The LogToDB alarm works on Windows only.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ ifEntry

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ IfLoadRates (poll)


Events Sent to OpenView: [Table 7-3](#) summarizes the events this behavior model sends to OpenView:

TABLE 7-3. Events Sent to OpenView by IfLoad

Event Name	Inform Number	Description
NC_Interface_Load	7003	The throughput for this interface has been high over a persistent period of time.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Set counters for persistence
See [Alarm Counter in Designing and Managing Behavior Models](#) for details. See [Defining a Poll in Designing and Managing Behavior Models](#) about setting the poll rate or [Fire Trigger in Designing and Managing Behavior Models](#) about setting a delay time for details.
- ◆ Poll conditions for device load thresholds
See [Writing a Poll Condition in Designing and Managing Behavior Models](#) for details.
- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIBus
See [Associating an Action with a Transition in Designing and Managing Behavior Models](#) and [Inform Platform in Designing and Managing Behavior Models](#) for details.
- ◆ Log To File alarm action options
(file pathname, log action variables, verbose or non-verbose output)
See [Associating an Action with a Transition in Designing and Managing Behavior Models](#) and [Log to File in Designing and Managing Behavior Models](#) for details.
- ◆ Log To Database alarm action options
(log identity number and log action variables)
See [Associating an Action with a Transition in Designing and Managing Behavior Models](#) and [Log to Database in Designing and Managing Behavior Models](#) for details.

IfUpDownStatus

Path: /model/interface_status/updown/ifupdownstatus.mod

This multi-alarm behavior model determines whether each interface on a managed node is up, down, or testing. If an interface is down, NerveCenter sends a 7002 Inform to the network management platform.

Two additional alarms, IfUpDownStatus_LogToDB and IfUpDownStatus_LogToFile, are part of this model. When you enable these alarms, they perform a Log To Database or a Log to File alarm action, respectively, on each transition so that reports on interface status can be produced.



NOTE

The LogToDB alarm works on Windows only.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ ifEntry

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ IfStatus (poll)

Events Sent to OpenView: [Table 7-4](#) summarizes the events this behavior model sends to OpenView:

TABLE 7-4. Events Sent to OpenView by IfUpDownStatus

Event Name	Inform Number	Description
NC_Interface_Down	7002	It has been determined that the interface is operationally down.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.



TIP

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIBus

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

- ◆ Log To File alarm action options
(file pathname, log action variables, verbose or non-verbose output)

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.

- ◆ Log To Database alarm action options
(log identity number and log action variables)

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to Database in Designing and Managing Behavior Models* for details.

IfUpDownStatusByType

Path: /model/interface_status/updown_bytype/ifupdownstatus_bytype.mod

IfUpDownStatusByType is one of the multi-alarm behavior models shipped with NerveCenter and provides interface management for devices that can be managed using the MIB-II and Frame Relay MIBs. This model manages the following types of interfaces:

- ◆ Asynchronous Transfer Mode (ATM)
- ◆ Integrated Services Digital Network (ISDN)
- ◆ Fiber Distributed Data Interface (FDDI)
- ◆ Frame Relay Permanent Virtual Circuit (PVC) subinterfaces
- ◆ Frame Relay
- ◆ Local Area Network (LAN)
- ◆ Switched Multimegabit Data Service (SMDS)
- ◆ Synchronous Optical Network (SONET)
- ◆ Wide Area Network (WAN)

The majority of the alarms in this model are subobject scope alarms that categorize an interface (the possible categories are listed above) and then monitor its status. For most interfaces, the interface can be up, down, or in testing mode. (The exception is a Frame Relay PVC, which can only be up or down.)

When an alarm instance transitions to one of these states, it executes an Inform action to notify OpenView Network Node Manager of the new state. For this Inform action to have the desired effect, you must integrate the trapd.conf.txt file supplied with these models with the standard NerveCenter trapd.conf.

The interface status alarms are listed below:

- ◆ IF-IfATMStatus
- ◆ IF-IfFDDIStatus
- ◆ IF-IfFramePVCStatus
- ◆ IF-IfFrameRelayStatus
- ◆ IF-IfISDNStatus
- ◆ IF-IfLANStatus
- ◆ IF-IfSMDSStatus

- ◆ IF-IfSonetStatus
- ◆ IF-IfWANStatus

The model file also includes three other alarms: IF-IfStatus, IF-IfColdWarmStart, and IF-IfNmDemand.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ frCircuitEntry (IF-IfFramePVCStatus) and ifEntry (all other alarms)

Trigger Generators:

Below is a list of the objects that generate triggers:

- ◆ If-ColdWarmStart (poll)
- ◆ If-IfAdminOp (poll)
- ◆ If-IfDown (poll)
- ◆ If-IfDown2 (poll)
- ◆ If-IfFramePVC (poll)
- ◆ If-IfFramePVCDown (poll)
- ◆ If-IfFramePVCUp (poll)
- ◆ If-IfNotAdminOp (poll)
- ◆ If-IfTesting (poll)
- ◆ If-IfTesting2 (poll)
- ◆ If-IfTypeNotAdminOp (poll)
- ◆ If-ColdStart (mask)
- ◆ If-FramePVCUpDown (mask)
- ◆ If-LinkDown (mask)
- ◆ If-LinkUp (mask)
- ◆ If-WarmStart (mask)
- ◆ IF-IfATMStatus (alarm)
- ◆ IF-IfColdWarmStart (alarm)
- ◆ IF-IfFDDISStatus (alarm)
- ◆ IF-IfFramePVCStatus (alarm)

- ◆ IF-IfFrameRelayStatus (alarm)
- ◆ IF-IfISDNStatus (alarm)
- ◆ IF-IfLANStatus (alarm)
- ◆ IF-IfNmDemand (alarm)
- ◆ IF-IfSMDSStatus (alarm)
- ◆ IF-IfSonetStatus (alarm)
- ◆ IF-IfStatus (alarm)
- ◆ If-IfWANStatus (alarm)
- ◆ IF-SelectType (Perl subroutine)

Events Sent to OpenView: *Table 7-5* summarizes the events this behavior model sends to OpenView:

TABLE 7-5. Events Sent to OpenView by IfUpDownStatusByType

Event Name	Inform Number	Description
NC_Interface_ColdWarmStart	1500	As a result of a device reboot, interface instances are no longer guaranteed to be valid. Thus, interface management has been restarted for this device.
NC_Interface_IfLanUp	1501	A LAN interface on this device has returned to an operational state.
NC_Interface_IfLanTesting	1502	A LAN interface desired to be operational on this device is in a testing state. No packets will be passed.
NC_Interface_IfLanDown	1503	A LAN interface desired to be operational on this device is in a down state.
NC_Interface_IfWanUp	1504	A WAN interface on this device has returned to an operational state.
EVENT NC_Interface_IfWanTesting	1505	A WAN interface desired to be operational on this device is in a testing state. No packets will be passed.
NC_Interface_IfWanDown	1506	A WAN interface desired to be operational on this device is in a down state.
NC_Interface_IfFrameRelayUp	1507	A Frame Relay interface on this device has returned to an operational state.

TABLE 7-5. Events Sent to OpenView by IfUpDownStatusByType (Continued)

Event Name	Inform Number	Description
NC_Interface_IfFrameRelayTesting	1508	A Frame Relay interface desired to be operational on this device is in a testing state. No packets will be passed.
NC_Interface_IfFrameRelayDown	1509	A Frame Relay interface desired to be operational on this device is in a down state.
NC_Interface_IfFrameRelayPVCUp	1510	A Frame Relay PVC on this device has returned to an operational state.
NC_Interface_IfFrameRelayPVCDown	1511	A Frame Relay PVC desired to be operational on this device is in a down state.
NC_Interface_IfATMUp	1512	An ATM interface on this device has returned to an operational state.
NC_Interface_IfATMTesting	1513	An ATM interface desired to be operational on this device is in a testing state. No packets will be passed.
NC_Interface_IfATMDown	1514	An ATM interface desired to be operational on this device is in a down state.
NC_Interface_IfFDDIUp	1515	A FDDI interface on this device has returned to an operational state.
NC_Interface_IfFDDITesting	1516	A FDDI interface desired to be operational on this device is in a testing state. No packets will be passed.
NC_Interface_IfFDDIDown	1517	A FDDI interface desired to be operational on this device is in a down state.
NC_Interface_IfSONETUp	1518	A SONET interface on this device has returned to an operational state.
NC_Interface_IfSONETTesting	1519	A SONET interface desired to be operational on this device is in a testing state. No packets will be passed.
NC_Interface_IfSONETDown	1520	A SONET interface desired to be operational on this device is in a down state.
NC_Interface_IfISDNUp	1521	A ISDN interface on this device has returned to an operational state.
NC_Interface_IfISDNTesting	1522	A ISDN interface desired to be operational on this device is in a testing state. No packets will be passed.

TABLE 7-5. Events Sent to OpenView by IfUpDownStatusByType (Continued)

Event Name	Inform Number	Description
NC_Interface_IfISDNDown	1523	A ISDN interface desired to be operational on this device is in a down state.
NC_Interface_IfSMDSUp	1524	A SMDS interface on this device has returned to an operational state.
NC_Interface_IfSMDSTesting	1525	A SMDS interface desired to be operational on this device is in a testing state. No packets will be passed.
NC_Interface_IfSMDSDown	1526	A SMDS interface desired to be operational on this device is in a down state.
NC_IntIfNnmDemand	1527	A NerveCenter initiated event which causes NNM to perform a nmdemandpoll on a particular node in order to force NNM to poll the node and correctly reflect the status of its interfaces.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

TIP

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Set timers for persistence

See *Defining a Poll in Designing and Managing Behavior Models* about setting the poll rate or *Fire Trigger in Designing and Managing Behavior Models* about setting a delay time for details.

- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIBus

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

- ◆ Log To File alarm action options
(file pathname, log action variables, verbose or non-verbose output)

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.

The node status models reside under `models/node_status`, and consist of:

Section	Description
NodeStatus on page 47	Monitors the status of nodes and, optionally, their SNMP agents.
NodeStatusDwnStrm on page 49	A downstream alarm suppression behavior model that uses node relationships to accurately monitor their status in a complex network.

NodeStatus

Path: `/model/node_status/nodestatus_models.mod`

This multi-alarm behavior model monitors the status of nodes and, optionally, their SNMP agents.

The `SnmpStatus` alarm monitors the status of nodes and their SNMP agents. Uses built-in triggers to determine if a device is up, down, or unreachable. Monitors nodes associated with a property group that contains the property system.

The `IcmpStatus` alarm monitors the status of nodes based on their response to ICMP echo requests (pings). Determines if a device is up, down, or unreachable. Nodes must be associated with a property group that contains the property `icmpStatus`, but they are not required to run SNMP agents.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ `system` (`SnmpStatus` alarm) and `icmpStatus` (`IcmpStatus` alarm)

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ `IS_IcmpFastPoll` (poll)
- ◆ `IS_IcmpPoll` (poll)
- ◆ `SnmpPoll` (poll)
- ◆ `SnmpFastPoll` (poll)

- ◆ SS_IcmpFastPoll (poll)
- ◆ SS_IcmpPoll (poll)
- ◆ ColdStart (mask)
- ◆ WarmStart (mask)
- ◆ SS_IcmpError (Perl subroutine)


Events Sent to OpenView: *Table 8-1* summarizes the events this behavior model sends to OpenView:

TABLE 8-1. Events Sent to OpenView by NodeStatus

Event Name	Inform Number	Description
NC_IcmpTimeout	1102	This inform event indicates that attempts to contact a node have failed due to ICMP timeout.
NC_SnmpStatus	1103	This inform event is triggered by the SNMP timeout and indicates that attempts to contact a node have failed.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Poll intervals
See *Defining a Poll in Designing and Managing Behavior Models* for details.
- ◆ Node suppression on/off
See *Defining a Poll in Designing and Managing Behavior Models* for details.
- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIBus
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

NodeStatusDwnStrm

Path: /model/node_status/nodestatus_dwnstrm.mod

NodeStatusDwnStrm is a downstream alarm suppression behavior model that monitors nodes in a complex network. Using topology information— from HP OpenView, or a file that you provide— this model uses the relationships between nodes to determine the status of those nodes accurately.

A multi-alarm behavior model, NodeStatusDwnStrm consists of the following alarms:

- ♦ **DwnStrmSnmStatus**—accurately monitors the status of nodes that have an SNMP agent by taking into consideration the status of the nodes' parents.
- ♦ **DwnStrmIcmpStatus**—accurately monitors the status of nodes that do not have an SNMP agent by taking into consideration the status of the nodes' parents.
- ♦ **DwnStrmSnmStatus_LogToDB** and **DwnStrmIcmpStatus_LogToDB**—perform a Log To Database alarm action on each transition so that reports on node status can be produced.



NOTE

The LogToDB alarms, work on Windows only.

For more information about NodeStatusDwnStrm, see *Downstream Alarm Suppression in Designing and Managing Behavior Models*.

Property: The following property must be in the property group for the nodes you want to monitor:

- ♦ system (SnmStatus alarm) and icmpStatus (IcmpStatus alarm)

Trigger Generators: Below is a list of the objects that generate triggers:

- ♦ IS_IcmpFastPoll (poll)
- ♦ IS_IcmpPoll (poll)
- ♦ SnmpPoll (poll)
- ♦ SnmpFastPoll (poll)
- ♦ SS_IcmpFastPoll (poll)
- ♦ SS_IcmpPoll (poll)
- ♦ ColdStart (mask)
- ♦ WarmStart (mask)
- ♦ SetNodeStatusDown (Perl subroutine)
- ♦ SetNodeStatusTesting (Perl subroutine)

- ◆ SetNodeStatusUnreachable (Perl subroutine)
- ◆ SetNodeStatusUp (Perl subroutine)
- ◆ SS_IcmpError (Perl subroutine)
- ◆ TestParentSetNode (Perl subroutine)
- ◆ TestParentStatus (Perl subroutine)


Events Sent to OpenView: The table below summarizes the events this model sends to OpenView:

TABLE 8-2. Events Sent to OpenView by NodeStatusDwnStrm

Event Name	Inform Number	Description
NC_IcmpTimeout	1102	This inform event indicates that attempts to contact a node have failed due to ICMP timeout.
NC_SnmpStatus	1103	This inform event is triggered by the SNMP timeout and indicates that attempts to contact a node have failed.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Poll intervals
See *Defining a Poll in Designing and Managing Behavior Models* for details.
- ◆ Node suppression on/off
See *Defining a Poll in Designing and Managing Behavior Models* for details.
- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIBus
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.
- ◆ Log To Database alarm action options (log identity number and log action variables)
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to Database in Designing and Managing Behavior Models* for details.

The TCP status models reside under models/tcp_status, and consist of:

Section	Description
<i>TcpConnMon on page 51</i>	Monitors the number of current TCP connections on each device and compares it with the maximum number.
<i>TcpRetransAlg on page 53</i>	Checks all devices that support the tcp MIB-II group and verifies that their transmission algorithm is the most efficient.
<i>TcpRetransMon on page 55</i>	Monitors the percentage of TCP retransmissions for a particular device.

TcpConnMon

Path: /model/tcp_status/tcp_conn_mon.mod

Monitors the number of current TCP connections on each device and compares it with the maximum number. Ignores devices that have dynamically-allocated maximum connections. Gives early warnings by detecting increasing levels of TCP connection saturation. When a device reaches 98 percent of the maximum TCP connections, NerveCenter sends a 7005 Inform to the platform.

Two additional alarms, TcpConnMon_LogToDB and TcpConnMon_LogToFile, are part of this model. When you enable these alarms, they perform a Log To Database or a Log to File alarm action, respectively, on each transition so that reports on TCP connections can be produced.



NOTE

The LogToDB alarm works on Windows only.

Property: The following property must be in the property group for the nodes you want to monitor:

- ♦ tcp

Trigger Generators: Below is a list of the objects that generate triggers:

- ♦ TcpConnectionMon (poll)

Events Sent to OpenView: *Table 9-1* summarizes the events this behavior model sends to OpenView:


TABLE 9-1. Events Sent to OpenView by TcpConnMon

Event Name	Inform Number	Description
NC_Tcp_Conn	7005	The current number of TCP connections is reaching a saturated state.

For complete information about events sent to OpenView, see the `trapd.conf.txt` file shipped with these behavior models residing in `/model/tcp_status`.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Poll intervals
See *Defining a Poll in Designing and Managing Behavior Models* for details.
- ◆ Poll conditions for TCP connection thresholds
See *Writing a Poll Condition in Designing and Managing Behavior Models* for details.
- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIBus
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.
- ◆ Log To File alarm action options (file pathname, log action variables, verbose or non-verbose output)
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.
- ◆ Log To Database alarm action options (log identity number and log action variables)
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to Database in Designing and Managing Behavior Models* for details.

TcpRetransAlg

Path: /model/tcp_status/tcp_retrans_alg.mod

Checks all devices that support the tcp MIB-II group and verifies that their transmission algorithm is the most efficient. An inefficient retransmission algorithm can severely affect the network efficiency of the device. If a device with an inefficient retransmission algorithm is found, NerveCenter sends a 7006 Inform to the platform.



NOTE

The most efficient TCP retransmission algorithm currently available is Van Jacobson's Algorithm.

Two additional alarms, TcpRetransAlg_LogToDB and TcpRetransAlg_LogToFile, are part of this model. When you enable these alarms, they perform a Log To Database or a Log to File alarm action, respectively, on each transition so that reports on TCP transmission algorithms can be produced.



NOTE

The LogToDB alarm works on Windows only.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ tcp

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ TcpRetransAlg (poll)

Events Sent to OpenView: [Table 9-2](#) summarizes the events this behavior model sends to OpenView:


TABLE 9-2. Events Sent to OpenView by TcpRetransAlg

Event Name	Inform Number	Description
NC_Tcp_RetAlg	7006	The current retransmission algorithm on this device is inefficient.

For complete information about events sent to OpenView, see the trapd.conf.txt file shipped with these behavior models residing in /model/tcp_status.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Poll intervals
See *Defining a Poll in Designing and Managing Behavior Models* for details.
- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIBus
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.
- ◆ Log To File alarm action options (file pathname, log action variables, verbose or non-verbose output)
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.
- ◆ Log To Database alarm action options (log identity number and log action variables)
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to Database in Designing and Managing Behavior Models* for details.

TcpRetransMon

Path: \model\tcp_status\tcp_retrans_mon.mod

Monitors the percentage of TCP retransmissions for a particular device. Retransmission occurs when messages use the TCP protocol and a message is not acknowledged. NerveCenter sends a 7007 Inform to the platform if a high number of retransmissions is detected.

Two additional alarms, TcpRetransMon_LogToDB and TcpRetransMon_LogToFile, are part of this model. When you enable these alarms, they perform a Log To Database or a Log to File alarm action, respectively, on each transition so that reports on TCP retransmissions can be produced.



NOTE

The LogToDB alarm works on Windows only.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ tcp

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ TcpRetransStatus (poll)

Events Sent to OpenView: [Table 9-3](#) summarizes the events this behavior model sends to OpenView:

TABLE 9-3. Events Sent to OpenView by TcpRetransMon

Event Name	Inform Number	Description
NC_Tcp_Retrans	7007	The current number of retransmissions is excessive for this device.

For complete information about events sent to OpenView, see the trapd.conf.txt file shipped with these behavior models residing in \model\tcp_status.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, see [Integrating NerveCenter with a Network Management Platform](#).

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.



TIP

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Poll intervals

See *Defining a Poll in Designing and Managing Behavior Models* for details.

- ◆ Poll conditions for TCP connection thresholds

See *Writing a Poll Condition in Designing and Managing Behavior Models* for details.

- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIBus

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

- ◆ Log To File alarm action options (file pathname, log action variables, verbose or non-verbose output)

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.

- ◆ Log To Database alarm action options (log identity number and log action variables)

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to Database in Designing and Managing Behavior Models* for details.

The troubleshooting models reside under models/troubleshooting, and consist of:

Section	Description
InformConnectStatus on page 57	Monitors the status of NerveCenter's connection with the OpenView Platform Adapter (OVPA) as well as the Inform Acknowledgement Queue.

InformConnectStatus

Path: /model/troubleshooting/informconnectstatus.mod

This behavior model monitors the status of NerveCenter's connection with the OpenView Platform Adapter (OVPA) as well as the Inform Acknowledgement Queue.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ NO_PROP


Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ InformConnectStatus (alarm)

Events Sent to OpenView: InformConnectStatus sends no events to OpenView.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Set timers for persistence

See *Defining a Poll in Designing and Managing Behavior Models* about setting the poll rate or *Fire Trigger in Designing and Managing Behavior Models* about setting a delay time for details.

- ◆ Log To File alarm action options (file pathname, log action variables, verbose or non-verbose output)

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.

This chapter presents vendor-specific models for:

- ◆ *Cisco (Catalyst Switch Models)*
- ◆ *Cisco (Environmental Monitoring Models)*
- ◆ *Cisco (Legacy Models)*
- ◆ *SNMP Research Models*
- ◆ *Wellfleet (Nortel) Models*

The vendor models reside under model/vendors, and consist of:

Section	Description
Cisco Catalyst switch models	Behavior models that monitor various attributes of Cisco switches
<i>CatalystAlarm on page 62</i>	Monitors Minor/Major Chassis alarms and notifies other alarms
<i>CatalystFanStatus on page 63</i>	Monitors the status of the Cisco Switch fan module
<i>CatalystModuleStatus on page 64</i>	Monitors the Cisco switch module status
<i>CatalystPs1Status on page 65</i>	Monitors the status of Power Supply 1 in the Cisco Switch
<i>CatalystPs2Status on page 65</i>	Monitors the status of Power Supply 2 in the Cisco Switch
<i>CatalystPsTestResult on page 65</i>	Monitors the Power Supply test result for the Cisco Switch
Cisco environmental models	Behavior models that monitor environmental status.
<i>CiscoConfigChange on page 66</i>	Monitors notification of runtime and startup config changes
<i>CiscoCpuUtilization on page 67</i>	Monitors CPU usage of a Cisco router
<i>CiscoFanStatus on page 68</i>	Monitors the status of the Cisco router fan module

Section	Description
<i>CiscoMemoryPoolStatus on page 69</i>	Monitors the memory pool usage of a Cisco router
<i>CiscoPwrSupplyStatus on page 70</i>	Monitors the of the Cisco router power supplies
<i>CiscoTempStatus on page 71</i>	Monitors a Cisco router for temperature issues
<i>CiscoVoltageStatus on page 72</i>	Monitors a Cisco router for voltage issues
Cisco router models	Behavior models that monitor various attributes of Cisco routers
<i>CiscoBufferUtilization on page 73</i>	Monitors the free element buffers on Cisco routers to determine if their buffer availability is at acceptable levels
<i>CiscoCPUUtilization on page 75</i>	Monitors the CPU utilization on Cisco routers.****
<i>CiscoCRCCheck on page 77</i>	Monitors the CRC errors on Cisco router interfaces
<i>CiscoFanCheck on page 79</i>	Monitors the fan status in a Cisco router. ****
<i>CiscoMemGetCheck on page 80</i>	Monitors the available free memory for Cisco routers.
<i>CiscoPacketsDrop on page 82</i>	Monitors packets on Cisco router interfaces. Also monitors when both inbound and outbound packets are being dropped during the poll periods
<i>CiscoPwrSupplyCheck on page 84</i>	Monitors the power supply status in a Cisco router. ****
<i>CiscoTempCheck on page 86</i>	Monitors the temperature status in a Cisco router. ****
Compaq device models	Behavior models that monitor various attributes of Compaq devices
<i>CpqDaLogDrvStat on page 88</i>	Detects logical drive status changes such as overheating, a bad connection, expanding, or a failure
<i>CpqNic on page 90</i>	Detects when a Network Interface Card (NIC) has failed on a Compaq server. CpqNic also detects when the server has switched to another redundant NIC
<i>CpqDaPhyDrvThresh on page 91</i>	Monitors the threshold of an IDA physical drive
<i>CpqScsiPhyStat on page 92</i>	Monitors the physical SCSI drive status

Section	Description
<i>CpqUnknown on page 94</i>	Logs all traps from Compaq agents that are not used in the CpqTraps trap mask
<i>CpqUps on page 94</i>	Detects the failure and degradation of a computer's uninterruptible power supply (UPS) system
SNMP Research SNMPv3 models	Behavior models that use SNMP Research's SNMPv3 entity, CIAgent to monitor system resources
<i>SnmPresCritAppStatus on page 97</i>	Provides status of applications monitored by SNMP Research's CIAgent
<i>SnmPresDiskUsage on page 99</i>	Provides status of file systems monitored by SNMP Research's CIAgent
<i>SnmPresGenericLogMonitor on page 101</i>	Monitors log file events reported by SNMP Research's CIAgent
<i>SnmPresLoginMonitor on page 103</i>	Monitors failed logins reported by SNMP Research's CIAgent
<i>SnmPresMemUsage on page 105</i>	Monitors status of system memory
<i>SnmPresProcessorLoad on page 106</i>	Monitors the processor load
Wellfleet (Nortel) router models	Behavior models that monitor various attributes of Wellfleet (Nortel) routers
<i>WfBufferUtilization on page 108</i>	Monitors the free packet buffers on Wellfleet (Nortel) routers to determine if their buffer availability is at acceptable levels
<i>WfFanCheck on page 110</i>	Monitors the fan status in a Wellfleet (Nortel) router
<i>WfMemCheck on page 111</i>	Monitors the available free memory for Wellfleet (Nortel) routers
<i>WfPwrSupplyCheck on page 113</i>	Monitors the power supplies status in a Wellfleet (Nortel) router
<i>WfTempCheck on page 115</i>	Monitors the temperature status in a Wellfleet (Nortel) router

Cisco (Catalyst Switch Models)

These models can be found under `/model/vendors/cisco/catalyst/`

- ◆ *CatalystAlarm*
- ◆ *CatalystFanStatus*
- ◆ *CatalystModuleStatus*
- ◆ *CatalystPs1Status, CatalystPs2Status, CatalystPsTestResult*

CatalystAlarm

Path: `model/vendors/cisco/catalyst /catalystalarm.mod`

This behavior model monitors minor and major chassis alarm traps received from a Cisco Switch. When one of these conditions is detected it is logged to the `CatChassAlarm.log` file and triggers are fired to notify the other alarms of the occurrence of an alarm.

A log to file exists on each transition.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ `chassisGrp`

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ `CatalystAlarmOffTrap (mask)`
- ◆ `CatalystAlarmOnTrap (mask)`

Events Sent to OpenView: Currently there are no events sent to OpenView from this model.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customizations: Below is a list of the behavior model setting that you might want to customize to suite the requirements of your site.

TIP

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

CatalystFanStatus

Path: model/vendors/cisco/catalyst/catalystfanstatus.mod

This behavior model monitors the status of the fan module of a Cisco switch. The model checks directly for trouble with the fan module as well as reacts to a chassis alarm if detected. Changes in status are logged to the CatChassFan.log file.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ chassisGrp

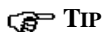
Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ CatalystFanStatusPoll (poll)
- ◆ CatalystFanStatusFastPoll (poll)
- ◆ CatalystAlarm (alarm)

Events Sent to OpenView: Currently there are no events sent to OpenView from this model.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customizations: Below is a list of the behavior model setting that you might want to customize to suite the requirements of your site.



TIP

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Add the appropriate platform inform actions to the FanTrouble transition

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

CatalystModuleStatus

Path: model/vendors/cisco/catalyst /catalystmodulestatus.mod

This behavior model monitors a cisco switch watching for module faults. It monitor the switch and detects the reception of a Module Slot Trap as well as Module fault and Module other conditions. When one of these conditions is detected it is logged to the CatModuleStatus.log file.

A log to file exists on each transition.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ moduleEntry


Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ CatalystModuleTrap (mask)
- ◆ CatalystModuleStatusPoll (poll)

Events Sent to OpenView: Currently there are no events sent to OpenView from this model.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customizations: Below is a list of the behavior model setting that you might want to customize to suite the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Add the appropriate platform inform actions to the ModuleFault transition

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

CatalystPs1Status, CatalystPs2Status, CatalystPsTestResult

Path: model/vendors/cisco/catalyst/catalystpsstatus.mod

These behavior models work together to monitor the redundant power supplies in a Cisco switch. The models check for Power Major and minor conditions for each power supply as well as monitoring the appropriate test result. Changes in status are logged to the CatChassPower.log file.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ chassisGrp

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ CatalystPs1StatusPoll (poll)
- ◆ CatalystPs1StatusFastPoll (poll)
- ◆ CatalystPs1TestResultPoll (poll)
- ◆ CatalystPs2StatusPoll (poll)
- ◆ CatalystPs2StatusFastPoll (poll)
- ◆ CatalystPs2TestResultPoll (poll)
- ◆ CatalystAlarm (alarm)
- ◆ CatalystPs1Status (alarm)
- ◆ CatalystPs2Status (alarm)
- ◆ CatalystPsTestResult (alarm)
- ◆ CatalystPsTestResult (Perl Subroutine)

Events Sent to OpenView: Currently there are no events sent to OpenView from this model.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customizations: Below is a list of the behavior model setting that you might want to customize to suite the requirements of your site.

TIP

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Add the appropriate platform inform actions to the PowerMinor and PowerMajor transitions

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

Cisco (Environmental Monitoring Models)

These models can be found under `/model/vendors/cisco/`

- ◆ *CiscoConfigChange*
- ◆ *CiscoCpuUtilization*
- ◆ *CiscoFanStatus*
- ◆ *CiscoMemoryPoolStatus*
- ◆ *CiscoPwrSupplyStatus*
- ◆ *CiscoTempStatus*
- ◆ *CiscoVoltageStatus*

CiscoConfigChange

Path: `model/vendors/cisco /ciscoconfigchange.mod`

This behavior model monitors Cisco Configuration changes which might occur during runtime or startup. Changes are logged to the `RouterConfigChange.log` file.

A log to file exists on each transition.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ `ccmHistory`


Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ `CiscoConfigChangePoll` (poll)
- ◆ `CiscoConfigChangeTrap` (mask)

Events Sent to OpenView: Currently there are no events sent to OpenView from this model.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Add an inform action to be notified of config changes when they occur

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

CiscoCpuUtilization

Path: model/vendors/cisco/ciscocpuutilization.mod

This behavior model monitors the CPU utilization of a Cisco router. The model checks for busy, very busy, and if sustained CPU critical conditions. Changes in status are logged to the ciscoCpuUtil.log file.

A log to file exists on each transition.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ cpmCPUTotalEntry


Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ CiscoCpuCheck (poll)
- ◆ CiscoCpuCheckFast (poll)

Events Sent to OpenView: Currently there are no events sent to OpenView from this model.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Add the appropriate platform inform actions to the ciscoCpuCritical transition
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.
- ◆ Adjust the thresholds in the trigger generators for the cpu busy and cpu very busy conditions
See *Writing a Poll Condition in Designing and Managing Behavior Models* for details.
- ◆ Adjust the timing on the transitions firetrigger used to verify the threshold violation over a sustained period of time

CiscoFanStatus

Path: model/vendors/cisco/cisconfanstatus.mod

This behavior model monitors the status of the fan module of a Cisco router. The model checks for fan not present, fan fault and fan shutdown conditions. Changes in status are logged to the ciscoFanStatus.log file.

A log to file exists on each transition.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ ciscoEnvMonFanStatusEntry

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ CiscoFanStatusPoll (poll)
- ◆ CiscoFanStatusTrap (mask)

Events Sent to OpenView: Currently there are no events sent to OpenView from this model.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

TIP

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Add the appropriate platform inform actions to the FanFault and FanShutdown transitions
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

CiscoMemoryPoolStatus

Path: model/vendors/cisco/ciscomemorypoolstatus.mod

This behavior model monitors the memory pool of a Cisco router. The model checks for the memory pool to be at a warning, critical, sustained warning and sustained critical level. Transitions to and from the sustained states are logged to the ciscoMemPoolStatus.log file.

A log to file exists on the transitions to the sustained states.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ ciscoMemoryPoolEntry

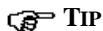
Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ CiscoMemPool (poll)

Events Sent to OpenView: Currently there are no events sent to OpenView from this model.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.



TIP

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Add the appropriate platform inform actions to the sustained transitions
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.
- ◆ Adjust the thresholds in the trigger generators for the warning and critical conditions
See *Fire Trigger in Designing and Managing Behavior Models* and *Writing a Poll Condition in Designing and Managing Behavior Models* for details.
- ◆ Adjust the alarm counters on the transitions used to verify the threshold violation over a sustained period of time
See *NC::AlarmCounters in Designing and Managing Behavior Models* and *Writing a Poll Condition in Designing and Managing Behavior Models* for details.

CiscoPwrSupplyStatus

Path: model/vendors/cisco/ciscopwrsupplystatus.mod

This behavior model monitors the status of the power supply of a Cisco router. The model checks for power supply not present, power supply fault and power supply shutdown conditions. Changes in status are logged to the ciscoPwrSupplyStatus.log file.

A log to file exists on each transition.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ ciscoEnvMonSupplyStatusEntry


Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ CiscoPwrSuppStatusPoll (poll)
- ◆ CiscoPwrSupplyStatusTrap (mask)

Events Sent to OpenView: Currently there are no events sent to OpenView from this model.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Add the appropriate platform inform actions to the PwrSupplyFault and PwrSupplyShut transitions

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

CiscoTempStatus

Path: model/vendors/cisco/ciscotempstatus.mod

This behavior model monitors the status of the temperature of a Cisco router. The model checks for temperature sensor not present, temperature fault and temperature shutdown conditions. Changes in status are logged to the ciscoTempCheck.log file.

A log to file exists on each transition.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ ciscoEnvMonTemperatureStatusEntry


Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ CiscoTempStatusPoll (poll)
- ◆ CiscoTempStatusTrap (mask)

Events Sent to OpenView: Currently there are no events sent to OpenView from this model.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Add the appropriate platform inform actions to the TempFault and TempShutdown transitions

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

CiscoVoltageStatus

Path: model/vendors/cisco/ciscovoltagestatus.mod

This behavior model monitors the voltage status of a Cisco router. The model checks for voltage sensor not present, voltage fault and voltage shutdown conditions. Changes in status are logged to the ciscoVoltStatus.log file.

A log to file exists on each transition.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ ciscoEnvMonVoltageStatusEntry


Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ CiscoVoltageStatusPoll (poll)
- ◆ CiscoVoltageStatusTrap (mask)

Events Sent to OpenView: Currently there are no events sent to OpenView from this model.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Add the appropriate platform inform actions to the VoltFault and VoltShutdown transitions. See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

Cisco (Legacy Models)

These models can be found under `/model/vendors/cisco/archive`

- ◆ *CiscoBufferUtilization*
- ◆ *CiscoCPUUtilization*
- ◆ *CiscoCRCCheck*
- ◆ *CiscoFanCheck*
- ◆ *CiscoMemGetCheck*
- ◆ *CiscoPacketsDrop*
- ◆ *CiscoPwrSupplyCheck*
- ◆ *CiscoTempCheck*

CiscoBufferUtilization

Path: `/model/vendors/cisco/archive/cisco_buffer_utilization.mod`

Monitors the free element buffers on Cisco routers to determine if their buffer availability is at acceptable levels. If buffer availability is less than five percent for three poll intervals, then NerveCenter sends a 6005 Inform to the platform.

Two additional alarms, `CiscoBufferUtilization_LogToDB` and `CiscoBufferUtilization_LogToFile`, are part of this model. When enabled, they perform a Log To Database or a Log to File alarm action, respectively, on each transition so that reports on buffer utilization can be produced.



NOTE

The `LogToDB` alarm works on Windows only.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ `lsystem`

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ `CiscoBuffersCheck` (poll)

Events Sent to OpenView: [Table 11-1](#) summarizes the events this model sends to OpenView:


TABLE 11-1. Events Sent to OpenView by CiscoBufferutilization

Event Name	Inform Number	Description
NC_Cisco_Buffer	6005	The buffer utilization for this device has been high over a persistent period of time.

For complete information about events sent to OpenView, see the `trapd.conf.txt` file shipped with these behavior models residing in `/model/vendors/cisco`.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Set timers for persistence
See *Defining a Poll in Designing and Managing Behavior Models* about setting the poll rate or *Fire Trigger in Designing and Managing Behavior Models* about setting a delay time for details.
- ◆ Poll intervals
See *Defining a Poll in Designing and Managing Behavior Models* for details.
- ◆ Poll conditions for buffer utilization thresholds
See *Writing a Poll Condition in Designing and Managing Behavior Models* for details.
- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIBus
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.
- ◆ Log To File alarm action options (file pathname, log action variables, (non-)verbose output)
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.
- ◆ Log To Database alarm action options (log identity number and log action variables)

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to Database in Designing and Managing Behavior Models* for details.

CiscoCPUUtilization

Path: /model/vendors/cisco/archive/cisco_cpu_utilization.mod

This behavior model monitors the CPU utilization on Cisco routers. NerveCenter sends a 6001 Inform to the platform only when the utilization reaches a very busy state and remains there for one hour.

Two additional alarms, CiscoCpuUtilization_LogToDB and CiscoCpuUtilization_LogToFile, are part of this model. When you enable these alarms, they perform a Log To Database or a Log to File alarm action, respectively, on each transition so that reports on CPU utilization can be produced.



NOTE

The LogToDB alarm works on Windows only.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ lsystem

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ CiscoCpuCheck (poll)
- ◆ CiscoCpuUtilization (alarm)
- ◆ CiscoCpuUtilization_LogToDB (alarm)
- ◆ CiscoCpuUtilization_LogToFile (alarm)

Events Sent to OpenView: *Table 11-2* summarizes the events this behavior model sends to OpenView:


TABLE 11-2. Events Sent to OpenView by CiscoCPUUtilization

Event Name	Inform Number	Description
NC_Cisco_Cpu	6001	The CPU utilization for this device has been high over a persistent period of time.

For complete information about events sent to OpenView, see the trapd.conf.txt file shipped with these behavior models residing in /model/vendors/cisco.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Set timers for persistence

See *Defining a Poll in Designing and Managing Behavior Models* about setting the poll rate or *Fire Trigger in Designing and Managing Behavior Models* about setting a delay time for details.

- ◆ Poll intervals

See *Defining a Poll in Designing and Managing Behavior Models* for details.

- ◆ Poll conditions for CPU utilization thresholds

See *Writing a Poll Condition in Designing and Managing Behavior Models* for details.

- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIbus

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

- ◆ Log To File alarm action options (file pathname, log action variables, verbose or non-verbose output)

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.

- ◆ Log To Database alarm action options (log identity number and log action variables)

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to Database in Designing and Managing Behavior Models* for details.

CiscoCRCCheck

Path: /model/vendors/cisco/archive/cisco_crc_check.mod

This model monitors the CRC errors on Cisco router interfaces. A poll checks to see if an increase in the number of CRC errors is occurring. If the threshold is exceeded, then a faster poll continues to check the interface for two more intervals. If the interface continues to exhibit CRC errors, NerveCenter sends a 6009 Inform to the platform.

Two additional alarms, CiscoCRCCheck_LogToDB and CiscoCRCCheck_LogToFile, are part of this model. When you enable these alarms, they perform a Log To Database or a Log to File alarm action, respectively, on each transition so that reports on CRC errors can be produced.



NOTE

The LogToDB alarm works on Windows only.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ lifEntry

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ CiscoCRCPoll (poll)
- ◆ CiscoCRCPollFast (poll)

Events Sent to OpenView: [Table 11-3](#) summarizes the events this behavior model sends to OpenView:


TABLE 11-3. Events Sent to OpenView by CiscoCRCCheck

Event Name	Inform Number	Description
NC_Cisco_CRC	6009	The amount of CRC errors for this interface has been high over a persistent period of time.

For complete information about events sent to OpenView, see the trapd.conf.txt file shipped with these behavior models residing in /model/vendors/cisco.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Set counters for persistence

See *Alarm Counter in Designing and Managing Behavior Models* for details. See *Defining a Poll in Designing and Managing Behavior Models* about setting the poll rate or *Fire Trigger in Designing and Managing Behavior Models* about setting a delay time for details.

- ◆ Poll intervals

See *Defining a Poll in Designing and Managing Behavior Models* for details.

- ◆ Poll conditions for CRC error thresholds

See *Writing a Poll Condition in Designing and Managing Behavior Models* for details.

- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIBus

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

- ◆ Log To File alarm action options (file pathname, log action variables, verbose or non-verbose output)

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.

- ◆ Log To Database alarm action options (log identity number and log action variables)

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to Database in Designing and Managing Behavior Models* for details.

CiscoFanCheck

Path: /model/vendors/cisco/archive/cisco_fan_check.mod

This behavior model monitors the fan status in a Cisco router. If a trap or a poll shows a problem, the alarm transitions to the FanWarning state which then polls the fan status again for verification. If the problem persists, NerveCenter sends a 6006 Inform to the platform. Once the problem has been corrected, the alarm returns to ground.

Two additional alarms, CiscoFanCheck_LogToDB and CiscoFanCheck_LogToFile, are part of this model. When you enable these alarms, they perform a Log To Database or a Log to File alarm action, respectively, on each transition so that reports on fan conditions can be produced.



NOTE

The LogToDB alarm works on Windows only.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ ciscoEnvMonFanStatusEntry

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ CiscoFanPoll (poll)
- ◆ CiscoFanPollFast (poll)
- ◆ CiscoFanFailureTrap (mask)

Events Sent to OpenView: [Table 11-4](#) summarizes the events this behavior model sends to OpenView:


TABLE 11-4. Events Sent to OpenView by CiscoFanCheck

Event Name	Inform Number	Description
NC_Cisco_Fan	6006	A fan failure has been detected for this device.

For complete information about events sent to OpenView, see the trapd.conf.txt file shipped with these behavior models residing in /model/vendors/cisco.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Poll intervals

See *Defining a Poll in Designing and Managing Behavior Models* for details.

- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIBus

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

- ◆ Log To File alarm action options (file pathname, log action variables, verbose or non-verbose output)

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.

- ◆ Log To Database alarm action options (log identity number and log action variables)

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to Database in Designing and Managing Behavior Models* for details.

CiscoMemGetCheck

Path: /model/vendors/cisco/archive/cisco_mem_get_check.mod

This multi-alarm model monitors the available free memory for Cisco routers.

The CiscoMemGet alarm obtains the memory installed in a Cisco router. The alarm will poll routers every six hours to ensure the memory configuration has not changed. It also listens for warm and cold starts which causes the router to be polled in case memory has been installed or removed.

The CiscoMemCheck alarm checks to see if the problem persists over three polling intervals. The free memory is tested for: low (6-10%), very low (< 6%), and zero-free-memory available conditions. On each state a poll is run, passing the resulting data through a Perl subroutine which uses the total memory installed in the router—along with the free memory just polled—to determine the percentage available.

Two additional alarms, CiscoMemCheck_LogToDB and CiscoMemCheck_LogToFile, are part of this model. When you enable these alarms, they perform a Log To Database or a Log to File alarm action, respectively, on each transition so that reports on memory utilization can be produced.

**NOTE**

The LogToDB alarm works on Windows only.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ chassis (CiscoMemGet) and lsystem (CiscoMemCheck)

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ CiscoMemCheck (poll)
- ◆ CiscoMemGet (poll)
- ◆ CiscoMemGetFast (poll)
- ◆ ColdStart (mask)
- ◆ WarmStart (mask)
- ◆ CiscoMemCheck (Perl subroutine)

Events Sent to OpenView: [Table 11-5](#) summarizes the events this behavior model sends to OpenView:


TABLE 11-5. Events Sent to OpenView by CiscoMemGetCheck

Event Name	Inform Number	Description
NC_Cisco_LoMem	6010	The free memory available for this device has been low over a persistent period of time.
NC_Cisco_VeryLoMem	6011	The free memory available for this device has been very low over a persistent period of time.
NC_Cisco_ZeroMem	6012	The free memory available for this device has been zero over a persistent period of time.

For complete information about events sent to OpenView, see the trapd.conf.txt file shipped with these behavior models residing in /model/vendors/cisco.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Set counters for persistence

See *Alarm Counter in Designing and Managing Behavior Models* for details. See *Defining a Poll in Designing and Managing Behavior Models* about setting the poll rate or *Fire Trigger in Designing and Managing Behavior Models* about setting a delay time for details.

- ◆ Poll intervals

See *Defining a Poll in Designing and Managing Behavior Models* for details.

- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIbus

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

- ◆ Log To File alarm action options (file pathname, log action variables, verbose or non-verbose output)

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.

- ◆ Log To Database alarm action options (log identity number and log action variables)

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to Database in Designing and Managing Behavior Models* for details.

CiscoPacketsDrop

Path: /model/vendors/cisco/archive/cisco_packets_drop.mod

Looks for a significant number of dropped inbound, outbound, or both types of packets on Cisco router interfaces. It checks to see if the problem persists over three polling intervals. Also monitors when both inbound and outbound packets are being dropped during the poll periods.

Two additional alarms, CiscoPacketsDrop_LogToDB and CiscoPacketsDrop_LogToFile, are part of this model. When you enable these alarms, they perform a Log To Database or a Log to File alarm action, respectively, on each transition so that reports on packet errors can be produced.

**NOTE**

The LogToDB alarm works on Windows only.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ lifEntry

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ CiscoPktsDropCheck (poll)

Events Sent to OpenView: *Table 11-6* summarizes the events this behavior model sends to OpenView:

TABLE 11-6. Events Sent to OpenView by CiscoPacketsDrop

Event Name	Inform Number	Description
NC_Cisco_InPkts	6002	The amount of dropped inbound packets for this interface has been high over a persistent period of time.
NC_Cisco_OutPkts	6003	The amount of dropped outbound packets for this interface has been high over a persistent period of time.
NC_Cisco_BothPkts	6004	The amount of dropped in and out bound packets for this interface has been high over a persistent period of time.

For complete information about events sent to OpenView, see the trapd.conf.txt file shipped with these behavior models residing in /model/vendors/cisco.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

**TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Set counters for persistence

See *Alarm Counter in Designing and Managing Behavior Models* for details. See *Defining a Poll in Designing and Managing Behavior Models* about setting the poll rate or *Fire Trigger in Designing and Managing Behavior Models* about setting a delay time for details.

- ◆ Poll intervals
See *Defining a Poll in Designing and Managing Behavior Models* for details.
- ◆ Poll conditions for input and output queue lengths
See *Writing a Poll Condition in Designing and Managing Behavior Models* for details.
- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIBus
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.
- ◆ Log To File alarm action options (file pathname, log action variables, verbose or non-verbose output)
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.
- ◆ Log To Database alarm action options (log identity number and log action variables)
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to Database in Designing and Managing Behavior Models* for details.

CiscoPwrSupplyCheck

Path: /model/vendors/cisco/archive/cisco_pwr_supply_check.mod

This behavior model monitors the power supply status in a Cisco router. If a trap or a poll shows a problem, the alarm transitions to the PowerSupplyWarning state which then polls the power supply status again for verification. If there is a problem, NerveCenter sends a 6007 Inform to the platform. Once the problem has been corrected, the alarm returns to ground.

Two additional alarms, CiscoPwrSupplyCheck_LogToDB and CiscoPwrSupplyCheck_LogToFile, are part of this model. When you enable these alarms, they perform a Log To Database or a Log to File alarm action, respectively, on each transition so that reports on power supply status can be produced.



NOTE

The LogToDB alarm works on Windows only.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ ciscoEnvMonSupplyStatusEntry

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ CiscoPwrSupplyPoll (poll)
- ◆ CiscoPwrSupplyPollFast (poll)
- ◆ CiscoPwrSupplyFailureTrap (mask)

Events Sent to OpenView: *Table 11-7* summarizes the events this behavior model sends to OpenView:


TABLE 11-7. Events Sent to OpenView by CiscoPwrSupplyCheck

Event Name	Inform Number	Description
NC_Cisco_PowerSupply	6007	A power supply failure has been detected for this device.

For complete information about events sent to OpenView, see the trapd.conf.txt file shipped with these behavior models residing in /model/vendors/cisco.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **Tip**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Poll intervals
See *Defining a Poll in Designing and Managing Behavior Models* for details.
- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIBus
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.
- ◆ Log To File alarm action options (file pathname, log action variables, verbose or non-verbose output)
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.
- ◆ Log To Database alarm action options (log identity number and log action variables)
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to Database in Designing and Managing Behavior Models* for details.

CiscoTempCheck

Path: /model/vendors/cisco/archive/cisco_temp_check.mod

This behavior model monitors the temperature status in a Cisco router. If a trap or a poll shows a problem, the alarm transitions to the TempWarning state which then polls the temperature status again for verification. If there is a problem, NerveCenter sends a 6008 Inform to the platform. Once the problem has been corrected, the alarm returns to ground.

Two additional alarms, CiscoTempCheck_LogToDB and CiscoTempCheck_LogToFile, are part of this model. When you enable these alarms, they perform a Log To Database or a Log to File alarm action, respectively, on each transition so that reports on temperature status can be produced.



NOTE

The LogToDB alarm works on Windows only.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ ciscoEnvMonTemperatureStatusEntry

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ CiscoTempPoll (poll)
- ◆ CiscoTempPollFast (poll)
- ◆ CiscoTempTrap (mask)

Events Sent to OpenView: [Table 11-8](#) summarizes the events this behavior model sends to OpenView:


TABLE 11-8. Events Sent to OpenView by CiscoTempCheck

Event Name	Inform Number	Description
NC_Cisco_Temp	6008	An over-temperature condition has been detected for this device.

For complete information about events sent to OpenView, see the trapd.conf.txt file shipped with these behavior models residing in /model/vendors/cisco.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Poll intervals
See *Defining a Poll in Designing and Managing Behavior Models* for details.
- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIBus
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.
- ◆ Log To File alarm action options (file pathname, log action variables, verbose or non-verbose output)
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.
- ◆ Log To Database alarm action options (log identity number and log action variables)
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to Database in Designing and Managing Behavior Models* for details.

Compaq Models

The following models can be found under `/model/vendors/compaq/`

- ◆ *CpqDaLogDrvStat*
- ◆ *CpqNic*
- ◆ *CpqDaPhyDrvThresh*
- ◆ *CpqScsiPhyStat*
- ◆ *CpqUnknown*
- ◆ *CpqUps*

CpqDaLogDrvStat

Path: `/model/vendors/compaq/compaq_logical_drive.mod`

This behavior model detects logical drive status changes such as overheating, a bad connection, expanding, or a failure.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ `cpqDaLogDrvEntry`

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ `CpqDaLogDrvBadConn (poll)`
- ◆ `CpqDaLogFailed (poll)`
- ◆ `CpqDaLogOK (poll)`
- ◆ `CpqTraps (mask)`

Events Sent to OpenView: [Table 11-9](#) summarizes the events this behavior model sends to OpenView:

TABLE 11-9. Events Sent to OpenView by CpqDalogDrvStat

Event Name	Inform Number	Description
NC_CpqLogOverheat	1208	This inform event is triggered by the CpqDaLogDrvStat alarm to indicate that a an overheat indication has been detected on the physical drive.

TABLE 11-9. Events Sent to OpenView by CpqDalogDrvStat (Continued)

Event Name	Inform Number	Description
NC_CpqLogShutDown	1209	This inform event is triggered by the CpqDaLogDrvStat alarm to indicate that a physical drive has shutdown.
NC_CpqLogFailed	1210	This inform event is triggered by the CpqDaLogDrvStat alarm to indicate that the log device has failed.
NC_CpqLogBadConn	1211	This inform event is triggered by the CpqDaLogDrvStat alarm to indicate that the connection to the data log device has been lost.
NC_CpqLogOK	1212	This inform event is triggered by the CpqDaLogDrvStat alarm to indicate that the failed physical drive has recovered.

For complete information about events sent to OpenView, see the `trapd.conf.txt` file shipped with these behavior models residing in `/model/vendors/compaq`.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

TIP

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ SMTP mail recipient

See *SMTP Mail in Designing and Managing Behavior Models* for details and *Specifying an SMTP Server for Mail Notification in Managing NerveCenter*.

- ◆ Poll intervals

See *Defining a Poll in Designing and Managing Behavior Models* for details.

- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIBus

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

CpqNic

Path: /model/vendors/compaq/compaq_nic.mod

This behavior model detects when a Network Interface Card (NIC) has failed on a Compaq server. It also detects when the server has switched to another redundant NIC. When a failure is detected, NerveCenter extracts from the varbind information which slot contains the failed card. NerveCenter then sends this information in an email as well as informs the network management platform.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ compaq

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ CpqTraps (mask)

Events Sent to OpenView: [Table 11-10](#) summarizes the events this behavior model sends to OpenView:

TABLE 11-10. Events Sent to OpenView by CpqNic

Event Name	Inform Number	Description
NC_CpqNicFailed	1203	This inform event is triggered by the CpqNic alarm to indicate that a network interface card has failed.
NC_CpqNicOK	1204	This inform event is triggered by the CpqNic alarm to indicate that network interface card has recovered.

For complete information about events sent to OpenView, see the trapd.conf.txt file shipped with these behavior models residing in /model/vendors/compaq.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

TIP

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ SMTP mail recipient

See *SMTP Mail in Designing and Managing Behavior Models* for details and *Specifying an SMTP Server for Mail Notification in Managing NerveCenter*.

- ◆ Poll intervals

See *Defining a Poll in Designing and Managing Behavior Models* for details.

- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIBus

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

CpqDaPhyDrvThresh

Path: /model/vendors/compaq/compaq_physical_drive.mod

This behavior model monitors the threshold of an IDA physical drive. When NerveCenter receives a threshold trap, it polls the various attributes that track whether the device has exceeded its threshold. If the device has in fact exceeded the threshold, then the alarm transitions to PreFailure. If not, a diagnostic test can be run on the unit.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ cpqDaPhyDrvThrEntry

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ CpqDaPhyDrvThresh (poll)
- ◆ CpqDaPhyDrvThresh2 (poll)
- ◆ CpqTraps (mask)

Events Sent to OpenView: [Table 11-11](#) summarizes the events this behavior model sends to OpenView:


TABLE 11-11. Events Sent to OpenView by CpqDaPhyDrvThresh

Event Name	Inform Number	Description
NC_CpqDaPhyDrvFail	1214	This inform event is triggered by the CpqDaPhyDrvThresh alarm to indicate that the threshold of the device has been exceeded.

For complete information about events sent to OpenView, see the `trapd.conf.txt` file shipped with these behavior models residing in `/model/vendors/compaq`.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ SMTP mail recipient

See *SMTP Mail in Designing and Managing Behavior Models* for details and *Specifying an SMTP Server for Mail Notification in Managing NerveCenter*.

- ◆ Poll intervals

See *Defining a Poll in Designing and Managing Behavior Models* for details.

- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIbus

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

CpqScsiPhyStat

Path: `/model/vendors/compaq/compaq_scsi_drive.mod`

This behavior model monitors the physical SCSI drive status. It checks to see if the device has failed, has a bad cable connection, or has been removed.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ `cpqScsiPhyDrvEntry`

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ `CpqScsiPhyBadConn` (poll)
- ◆ `CpqScsiPhyFailed` (poll)
- ◆ `CpqScsiPhyOK` (poll)
- ◆ `CpqTraps` (mask)

Events Sent to OpenView: *Table 11-12* summarizes the events this behavior model sends to OpenView:

TABLE 11-12. Events Sent to OpenView by CpqSesiPhyStat

Event Name	Inform Number	Description
NC_CpqPhyOK	1205	This inform event is triggered by the CpqNic alarm to indicate that the physical SCSI drive has recovered.
NC_CpqPhyFail	1206	The inform event is triggered by the CpqNic alarm to indicate that the physical SCSI drive has recovered.
NC_CpqPhyBadConn	1207	This inform event is triggered by the CpqPhyBadConn alarm to indicate that a bad connection has been detected on the physical SCSI drive.

For complete information about events sent to OpenView, see the `trapd.conf.txt` file shipped with these behavior models residing in `/model/vendors/compaq`.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

Tip

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ SMTP mail recipient

See *SMTP Mail in Designing and Managing Behavior Models* for details and *Specifying an SMTP Server for Mail Notification in Managing NerveCenter*.

- ◆ Log file pathname

See *Log to File in Designing and Managing Behavior Models* for details.

- ◆ Poll intervals

See *Defining a Poll in Designing and Managing Behavior Models* for details.

- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIBus

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

CpqUnknown

Path: /model/vendors/compaq/compaq_unknown_traps.mod

This behavior model logs all traps from Compaq agents that are not used in the CpqTraps trap mask. This will allow the user to see which Compaq traps could be used in future behavior models.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ compaq

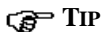
Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ CpqTraps (mask)

Events Sent to OpenView: CpqUnknown sends no events to OpenView.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.



TIP

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Log To File alarm action options (file pathname, log action variables, verbose or non-verbose output)

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.

CpqUps

Path: /model/vendors/compaq/compaq_ups.mod

This behavior model detects the failure and degradation of a computer's uninterruptible power supply (UPS) system. If a failure is detected via a trap, NerveCenter first takes a quick poll to make sure it is not a power surge. If it is not a surge, the alarm transitions in to the ACLineFail state. An e-mail is sent notifying the administrator that the UPS has lost AC power and how much time is left in the UPS. It will also include whether or not the system attached to the UPS will perform an automatic shutdown when the UPS is low on power.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ cpqUpsBasic

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ CpqUpsLowPower (poll)
- ◆ CpqUpsOK (poll)
- ◆ CpqUpsOKQuick (poll)
- ◆ CpqTraps (mask)
- ◆ CpqUps (alarm)

Events Sent to OpenView: *Table 11-13* summarizes the events this behavior model sends to OpenView:


TABLE 11-13. Events Sent to OpenView by CpqUps

Event Name	Inform Number	Description
NC_CpqUpsFail2	1200	This inform event is triggered by the CpqUps alarm to indicate that the AC power to the interruptible power supply has failed.
NC_CpqUpsOK	1201	This inform event is triggered by the CpqUps alarm to indicate that the AC power to the interruptible power supply has been restored.
NC_CpqUpsShutDown	1202	This inform event is triggered by the CpqUps alarm to indicate that the interruptible power supply has been shutdown.
NC_CpqUpsLow	1213	This inform event is triggered by the CpqUps alarm to indicate that a low battery level on the interruptible power supply has been shut down.

For complete information about events sent to OpenView, see the trapd.conf.txt file shipped with these behavior models residing in /model/vendors/compaq.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Set timers for persistence

See *Defining a Poll in Designing and Managing Behavior Models* about setting the poll rate or *Fire Trigger in Designing and Managing Behavior Models* about setting a delay time for details.

- ◆ Poll intervals

See *Defining a Poll in Designing and Managing Behavior Models* for details.

- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIBus

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

SNMP Research Models

These models can be found under `/model/vendors/snmp_research/`

- ◆ *SnmPRESCritAppStatus*
- ◆ *SnmPRESDiskUsage*
- ◆ *SnmPRESGenericLogMonitor*
- ◆ *SnmPRESLoginMonitor*
- ◆ *SnmPRESMemUsage*
- ◆ *SnmPRESProcessorLoad*

SnmPRESCritAppStatus

Path: `/model/vendors/snmp_research/snmpres_crit_app_status.mod`

This behavior model provides the status of the applications monitored by SNMP Research's CIAgent. When an application goes down and remains down for a specified time (the default is ten minutes), NerveCenter sends a 110031 Inform to the platform. When an application comes back up, the alarm returns to ground and sends a 120031 Inform to the platform. The nodes hosting the monitored applications must be associated with a property group that contains the property `critAppProcEntry`.

Property: The following property must be in the property group for the nodes hosting the applications you want to monitor:

- ◆ `critAppProcEntry`

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ `CritAppDown` (mask or poll)
- ◆ `CritAppUp` (mask or poll)
- ◆ `CritAppDownFast` (poll)
- ◆ `CritAppUpFast` (poll)
- ◆ `CritAppAdminDownFast` (poll)
- ◆ `CritAppDownNotify` (alarm)

Events Sent to OpenView: *Table 11-14* summarizes the events this behavior model sends to OpenView:


TABLE 11-14. Events Sent to OpenView by SnmpresCritAppStatus

Event Name	Inform Number	Description
NC_Crit_App_Down	110031	CritAppDown trap received from CIAgent, or polling determined that the application is down.
NC_Crit_App_Up	120031	CritAppUp trap received from CIAgent, or polling determined that the application is up.

For complete information about events sent to OpenView, see the `trapd.conf.txt` file shipped with these behavior models residing in `/model/vendors/snmp_research`.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Set timers for persistence

See *Defining a Poll in Designing and Managing Behavior Models* about setting the poll rate or *Fire Trigger in Designing and Managing Behavior Models* about setting a delay time for details.

- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIbus

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

CIAgent Customization: The following is a list of sample CIAgent configurations that you might want to customize to suit the requirements of your site:

- ◆ On Windows, the configuration file for the critical applications agent is:

```
C:\etc\srconf\agt\critagt.cnf
```

Below is a sample entry for monitoring the DNS service:

```
critAppProcEntry 2 DNS "net start dns" "net stop dns" 1 true 3000
true false \ true
```

- ◆ On UNIX, the configuration file for the critical applications agent is:

```
/etc/srconf/agt/critagt.cnf
```

Below is a sample entry for monitoring sendmail:

```
critAppProcEntry 1 sendmail "/etc/rc2.d/S88sendmail start"
"/etc/rc2.d/S88sendmail stop" 1 \
true 3000 true true true
```

The *true* parameter in these examples tells CIAgent to restart the application if it fails. Make this parameter false when you do not want CIAgent to restart the application.

SnmPRESDiskUsage

Path: /model/vendors/snmp_research/snmPRES_disk_usage.mod

This behavior model informs the status of the file systems monitored by SNMP Research's CIAgent. When the disk usage of a file system exceeds a threshold (90 percent by default) and remains above the threshold for a specified time (the default is fifteen minutes), NerveCenter sends a 110041 Inform to the platform. When the disk usage falls below the threshold, the alarm returns to ground and sends a 120041 Inform to the platform. The nodes that have their file systems monitored must be associated with a property group that contains the property siFsEntry.

Property: The following property must be in the property group for the nodes hosting the applications you want to monitor:

- ◆ siFsEntry

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ DiskUsageOk (mask or poll)
- ◆ HighDiskUsage (mask or poll)
- ◆ DiskUsageOkFast (poll)
- ◆ HighDiskUsageFast (poll)
- ◆ DiskUsageNotify (alarm)

Events Sent to OpenView: *Table 11-15* summarizes the events this behavior model sends to OpenView:

TABLE 11-15. Events Sent to OpenView by SnmpresDiskUsage

Event Name	Inform Number	Description
NC_Disk_Usage_High	110041	Disk usage over 90 percent.
NC_Disk_Usage_Ok	120041	Disk usage dropped under 90 percent.

For complete information about events sent to OpenView, see the `trapd.conf.txt` file shipped with these behavior models residing in `/model/vendors/snmp_research`.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

TIP

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Set timers for persistence

See *Defining a Poll in Designing and Managing Behavior Models* about setting the poll rate or *Fire Trigger in Designing and Managing Behavior Models* about setting a delay time for details.

- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIBus

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

SnmPRESGenericLogMonitor

Path: /model/vendors/snmp_research/snmpres_generic_log_monitor.mod

This behavior model monitors log file events reported by SNMP Research's CIAgent. When the amount of events reaches the threshold (three by default) within a specified time (the default is fifteen minutes), NerveCenter sends a 110081 Inform to the platform. If the time expires before the threshold is reached, the alarm is grounded and inform 120081 is sent. The nodes monitored must be associated with a property group that contains the property siLogEntry.

Property: The following property must be in the property group for the nodes hosting the applications you want to monitor:

- ◆ siLogEntry

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ LogFileEvent (mask)
- ◆ LogFileEventWarning (alarm)
- ◆ LogFileEventTimer (alarm)

Events Sent to OpenView: [Table 11-16](#) summarizes the events this behavior model sends to OpenView:


TABLE 11-16. Events Sent to OpenView by SnmpRESGenericLogMonitor

Event Name	Inform Number	Description
NC_Log_File_Event	110081	A logged event occurred three times in 15 minutes.
NC_Log_File_Event	120081	A logged event occurred less than three times in 15 minutes.

For complete information about events sent to OpenView, see the trapd.conf.txt file shipped with these behavior models residing in /model/vendors/snmp_research.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Set timers and counters for persistence

See *Alarm Counter in Designing and Managing Behavior Models* for details. See *Defining a Poll in Designing and Managing Behavior Models* about setting the poll rate or *Fire Trigger in Designing and Managing Behavior Models* about setting a delay time for details.

- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIbus

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

- ◆ Configure the mask to instantiate different alarms according to the log entry received

See *Using Trap Masks in Designing and Managing Behavior Models* for details.

CIAgent Customization: The following is a list of sample CIAgent configurations that you might want to customize to suit the requirements of your site:

- ◆ On Windows, there are three log files: Application, Security, and System. In order to monitor entries within them, you must not use the path to the file in the configuration entry. Instead, use “Event Log: *log filename: source*”, where *log filename* is the name of the log file; that is, Security, and the *source* is the application or service generating the log entry. Below are two sample entries; one matches a “failed to renew” string in a Dhcp event in the System log; the other matches “application error” strings in DrWatson events in the Application log:

```
siLogEntry 1 System "Event Log: System: Dhcp" "failed to renew" 2 4
0 1 - 1 \ 2031616 0 - 2 10 2 System 1
siLogEntry 2 Application "Event Log: Application: DrWatson"
"application error" 2 4 0 \ 0 - 1 524288 0 - 2 15 2 Application 1
```

The default path to the CIAgent log file monitoring configuration file on Windows is:

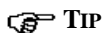
```
C:\etc\srconf\agt\logagt.cnf
```

- ◆ On UNIX, the location of the log files varies for every platform and application. Below is a sample entry for monitoring the syslog files on Solaris:

```
siLogEntry 1 Syslog /var/adm/messages "login: *pam_authenticate:
*error" 2 10 \ 200 25 - 1 8233138 0 - 2 15 2 System 1
```

The default path to the CIAgent log file monitoring configuration file on UNIX is:

```
/etc/srconf/agt/logagt.cnf
```



TIP

Verify that the appropriate message level for the events you are trying to monitor is configured in the `/etc/syslog.conf` file.

SnmPRESLoginMonitor

Path: `/model/vendors/snmp_research/snmPRES_login_monitor.mod`

This behavior model monitors failed logins reported by SNMP Research's CIAgent. When the number of attempts reaches the threshold (three by default) within a specified time (the default is five minutes), NerveCenter sends a 110071 Inform to the platform. If the time expires before the attempt threshold is reached, the alarm is grounded. The nodes monitored must be associated with a property group that contains the property `siLogEntry`.

Property: The following property must be in the property group for the nodes hosting the applications you want to monitor:

- ◆ `siLogEntry`

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ FailedLogin (mask)
- ◆ LoginWarning (alarm)
- ◆ LoginWarningTimer (alarm)

Events Sent to OpenView: [Table 11-17](#) summarizes the events this model sends to OpenView:


TABLE 11-17. Events Sent to OpenView by SnmPRESLoginMonitor

Event Name	Inform Number	Description
NC_Login_Failed	110071	Login attempted three times in five minutes.

For complete information about events sent to OpenView, see the `trapd.conf.txt` file shipped with these behavior models residing in `/model/vendors/snmp_research`.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Set timers and counters for persistence

See *Alarm Counter in Designing and Managing Behavior Models* for details. See *Defining a Poll in Designing and Managing Behavior Models* about setting the poll rate or *Fire Trigger in Designing and Managing Behavior Models* about setting a delay time for details.

- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIBus

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

- ◆ Return alarm to ground according to your security policy, after a timer expires or manually

See *Using Alarms in Designing and Managing Behavior Models* for details.


CIAGENT Customization: The following is a sample CIAgent configuration that you might want to customize to suit the requirements of your site:

On UNIX, the location of the log files varies for every platform and application. Below is a sample entry for monitoring the syslog file on Solaris:

```
siLogEntry 1 Syslog /var/adm/messages "login: *pam_authenticate:
*error" 2 10 \ 200 25 - 1 8233138 0 - 2 15 2 System 1
```

The default path to the CIAgent log file monitoring configuration file on UNIX is:

```
/etc/srconf/agt/logagt.cnf
```

 **TIP**

Verify that the appropriate message level for the events you are trying to monitor is configured in the `/etc/syslog.conf` file.

SnmpresMemUsage

Path: /model/vendors/snmp_research/snmpres_mem_usage.mod

This behavior model monitors the status of the system memory. When the memory utilization exceeds the threshold (90 percent by default) and remains above the threshold for a specified time (the default is fifteen minutes), NerveCenter sends a 110051 Inform to the platform. When the memory utilization falls below the threshold, the alarm returns to ground and sends a 120051 Inform to the platform. The nodes that have their memory utilization monitored must be associated with a property group that contains the property hrStorageEntry.

Property: The following property must be in the property group for the nodes hosting the applications you want to monitor:

- ◆ hrStorageEntry

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ MemUsageOk (poll)
- ◆ HighMemUsage (poll)
- ◆ MemUsageOkFast (poll)
- ◆ HighMemUsageFast (poll)
- ◆ MemUsageNotify (alarm)

Events Sent to OpenView: [Table 11-18](#) summarizes the events this behavior model sends to OpenView:


TABLE 11-18. Events Sent to OpenView by SnmpresMemUsage

Event Name	Inform Number	Description
NC_Mem_Usage_High	110051	Memory utilization exceeded 90 percent.
NC_Mem_Usage_Ok	120051	Memory utilization dropped under 90 percent.

For complete information about events sent to OpenView, see the trapd.conf.txt file shipped with these behavior models residing in /model/vendors/snmp_research.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Set timers for persistence

See *Defining a Poll in Designing and Managing Behavior Models* about setting the poll rate or *Fire Trigger in Designing and Managing Behavior Models* about setting a delay time for details.

- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIbus

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

SnmPRESProcessorLoad

Path: /model/vendors/snmp_research/snmpres_processor_load.mod

This behavior model monitors the processor load. When the processor load reaches the threshold (90 percent by default) and remains above the threshold for a specified time (the default is fifteen minutes), NerveCenter sends a 110061 Inform to the platform. When the processor load falls below the threshold, the alarm returns to ground and sends a 120061 Inform to the platform. The nodes that have their processors monitored must be associated with a property group that contains the property hrProcessorEntry.

Property:

The following property must be in the property group for the nodes hosting the applications you want to monitor:

- ◆ hrProcessorEntry

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ CPUok (poll)
- ◆ CPUokFast (poll)
- ◆ CPUbusy (poll)
- ◆ CPUbusyFast (poll)
- ◆ CPUbusyNotify (alarm)

Events Sent to OpenView: *Table 11-19* summarizes the events this behavior model sends to OpenView:


TABLE 11-19. Events Sent to OpenView by SnmpresProcessorLoad

Event Name	Inform Number	Description
NC_CPU_Busy	110061	Processor load reached 90 percent.
NC_CPU_Ok	120061	Processor load dropped under 90 percent.

For complete information about events sent to OpenView, see the `trapd.conf.txt` file shipped with these behavior models residing in `/model/vendors/snmp_research`.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Set timers for persistence

See *Defining a Poll in Designing and Managing Behavior Models* about setting the poll rate or *Fire Trigger in Designing and Managing Behavior Models* about setting a delay time for details.

- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIBus

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

Wellfleet (Nortel) Models

These models can be found under `/model/vendors/wellfleet/`

- ◆ *WfBufferUtilization*
- ◆ *WfFanCheck*
- ◆ *WfMemCheck*
- ◆ *WfPwrSupplyCheck*
- ◆ *WfTempCheck*

WfBufferUtilization

Path: `/model/vendors/wellfleet/wf_buffer_utilization.mod`

Monitors the free packet buffers on Wellfleet (Nortel) routers to determine if their buffer availability is at acceptable levels. If buffer availability is less than five percent for three poll intervals, then NerveCenter sends a 6505 Inform to the platform.

Two additional alarms, `WfBufferUtilization_LogToDB` and `WfBufferUtilization_LogToFile`, are part of this model. When you enable these alarms, they perform a Log To Database or a Log to File alarm action, respectively, on each transition so that reports on buffer availability can be produced.



NOTE

The `LogToDB` alarm works on Windows only.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ `wfKernelEntry`

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ `WfBuffersCheck` (poll)

Events Sent to OpenView: [Table 11-20](#) summarizes the events this behavior model sends to OpenView:

TABLE 11-20. Events Sent to OpenView by `WfBufferUtilization`

Event Name	Inform Number	Description
<code>NC_Wellfleet_Buffer</code>	6505	The buffer utilization for this device has been high over a persistent period of time.

For complete information about events sent to OpenView, see the `trapd.conf.txt` file shipped with these behavior models residing in `/model/vendors/wellfleet`.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization:

Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

TIP

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Set counters for persistence

See *Alarm Counter in Designing and Managing Behavior Models* for details. See *Defining a Poll in Designing and Managing Behavior Models* about setting the poll rate or *Fire Trigger in Designing and Managing Behavior Models* about setting a delay time for details.

- ◆ Poll intervals

See *Defining a Poll in Designing and Managing Behavior Models* for details.

- ◆ Poll conditions for buffer utilization thresholds

See *Writing a Poll Condition in Designing and Managing Behavior Models* for details.

- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIbus

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

- ◆ Log To File alarm action options (file pathname, log action variables, verbose or non-verbose output)

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.

- ◆ Log To Database alarm action options (log identity number and log action variables)

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to Database in Designing and Managing Behavior Models* for details.

WfFanCheck

Path: /model/vendors/wellfleet/wf_fan_check.mod

Monitors the fan status in a Wellfleet (Nortel) router. If a poll shows a problem, the alarm transitions to the FanWarning state which then polls the fan status again for verification. If there is a problem, NerveCenter sends a 6506 Inform to the platform. Once the problem has been corrected, the alarm returns to ground.

Two additional alarms, WfFanCheck_LogToDB and WfFanCheck_LogToFile, are part of this model. When you enable these alarms, they perform a Log To Database or a Log to File alarm action, respectively, on each transition so that reports on fan status can be produced.



NOTE

The LogToDB alarm works on Windows only.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ wfKernelEntry

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ WfFanPoll (poll)
- ◆ WfFanPollFast (poll)

Events Sent to OpenView: [Table 11-21](#) summarizes the events this behavior model sends to OpenView:


TABLE 11-21. Events Sent to OpenView by WfFanCheck

Event Name	Inform Number	Description
NC_Wellfleet_Fan	6506	A fan failure has been detected for this device.

For complete information about events sent to OpenView, see the trapd.conf.txt file shipped with these behavior models residing in /model/vendors/wellfleet.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Poll intervals
See *Defining a Poll in Designing and Managing Behavior Models* for details.
- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIbus
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.
- ◆ Log To File alarm action options (file pathname, log action variables, verbose or non-verbose output)
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.
- ◆ Log To Database alarm action options (log identity number and log action variables)
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to Database in Designing and Managing Behavior Models* for details.

WfMemCheck

Path: /model/vendors/wellfleet/wf_mem_check.mod

This model monitors the available free memory for Wellfleet (Nortel) routers. NerveCenter checks to see if the conditions persist over three polling intervals. The free memory is tested for the following conditions: low (6-10%), very low (< 6%), and zero-free-memory available.

Two additional alarms, WfMemCheck_LogToDB and WfMemCheck_LogToFile, are part of this model. When you enable these alarms, they perform a Log To Database or a Log to File alarm action, respectively, on each transition so that reports on memory availability can be produced.



NOTE

The LogToDB alarm works on Windows only.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ wfKernelEntry

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ WfMemCheck (poll)

Events Sent to OpenView: [Table 11-22](#) summarizes the events this behavior model sends to OpenView:

TABLE 11-22. Events Sent to OpenView by WfMemCheck

Event Name	Inform Number	Description
NC_Wellfleet_LoMem	6510	The free memory available for this device has been low over a persistent period of time.
NC_Wellfleet_VeryLoMem	6511	The free memory available for this device has been very low over a persistent period of time.
NC_Wellfleet_ZeroMem	6512	The free memory available for this device has been zero over a persistent period of time.

For complete information about events sent to OpenView, see the `trapd.conf.txt` file shipped with these behavior models residing in `/model/vendors/wellfleet`.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

TIP

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Set counters for persistence

See *Alarm Counter in Designing and Managing Behavior Models* for details. See *Defining a Poll in Designing and Managing Behavior Models* about setting the poll rate or *Fire Trigger in Designing and Managing Behavior Models* about setting a delay time for details.

- ◆ Poll intervals

See *Defining a Poll in Designing and Managing Behavior Models* for details.

- ◆ Poll conditions for memory utilization thresholds
See *Writing a Poll Condition in Designing and Managing Behavior Models* for details.
- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIbus
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.
- ◆ Log To File alarm action options (file pathname, log action variables, verbose or non-verbose output)
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.
- ◆ Log To Database alarm action options (log identity number and log action variables)
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to Database in Designing and Managing Behavior Models* for details.

WfPwrSupplyCheck

Path: /model/vendors/wellfleet/wf_pwr_supply_check.mod

This behavior model monitors the power supplies status in a Wellfleet (Nortel) router. If a poll shows a problem with any of the power supplies, the alarm transitions to the PowerSupplyWarning state which then polls the power supply status again for verification. If there is a problem, NerveCenter sends a 6507 Inform to the platform. Once the problem has been corrected, the alarm returns to ground.

Two additional alarms, WfPwrSupplyCheck_LogToDB and WfPwrSupplyCheck_LogToFile, are part of this model. When you enable these alarms, they perform a Log To Database or a Log to File alarm action, respectively, on each transition so that reports on power supply status can be produced.



NOTE

The LogToDB alarm works on Windows only.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ wfHwBase

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ WfPwrSupplyPoll (poll)
- ◆ WfPwrSupplyPollFast (poll)

Events Sent to OpenView: *Table 11-23* summarizes the events this behavior model sends to OpenView:


TABLE 11-23. Events Sent to OpenView by WfPwrSupplyCheck

Event Name	Inform Number	Description
NC_Wellfleet_PowerSupply	6507	A power supply failure has been detected for this device.

For complete information about events sent to OpenView, see the `trapd.conf.txt` file shipped with these behavior models residing in `/model/vendors/wellfleet`.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Poll intervals

See *Defining a Poll in Designing and Managing Behavior Models* for details.

- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIBus

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.

- ◆ Log To File alarm action options (file pathname, log action variables, verbose or non-verbose output)

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.

- ◆ Log To Database alarm action options (log identity number and log action variables)

See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to Database in Designing and Managing Behavior Models* for details.

WfTempCheck

Path: /model/vendors/wellfleet/wf_temp_check.mod

This behavior model monitors the temperature status in a Wellfleet (Nortel) router. If a poll shows a problem, the alarm transitions to the TempWarning state which then polls the temperature status again for verification. If there is a problem, NerveCenter sends a 6508 Inform to the platform. Once the problem has been corrected, the alarm returns to ground.

Two additional alarms, WfTempCheck_LogToDB and WfTempCheck_LogToFile, are part of this model. When you enable these alarms, they perform a Log To Database or a Log to File alarm action, respectively, on each transition so that reports on temperature conditions can be produced.



NOTE

The LogToDB alarm works on Windows only.

Property: The following property must be in the property group for the nodes you want to monitor:

- ◆ wfKernelEntry

Trigger Generators: Below is a list of the objects that generate triggers:

- ◆ WfTempPoll (poll)
- ◆ WfTempPollFast (poll)

Events Sent to OpenView: [Table 11-24](#) summarizes the events this behavior model sends to OpenView:


TABLE 11-24. Events Sent to OpenView by WfTempCheck

Event Name	Inform Number	Description
NC_Wellfleet_Temp	6508	An over-temperature condition has been detected for this device.

For complete information about events sent to OpenView, see the trapd.conf.txt file shipped with these behavior models residing in /model/vendors/wellfleet.

For more information about integrating NerveCenter with OpenView, or the other NerveCenter-supported network manager platforms, refer to the book *Integrating NerveCenter with a Network Management Platform*.

Optional Customization: Below is a list of behavior model settings that you might want to customize to suit the requirements of your site.

 **TIP**

It's a good practice to rename the behavior model objects that you modify. NerveCenter overwrites objects with the same name whenever you import a model.

- ◆ Poll intervals
See *Defining a Poll in Designing and Managing Behavior Models* for details.
- ◆ Change Inform to Inform Platform for IBM Tivoli Netcool/OMNIBus
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Inform Platform in Designing and Managing Behavior Models* for details.
- ◆ Log To File alarm action options (file pathname, log action variables, verbose or non-verbose output)
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to File in Designing and Managing Behavior Models* for details.
- ◆ Log To Database alarm action options (log identity number and log action variables)
See *Associating an Action with a Transition in Designing and Managing Behavior Models* and *Log to Database in Designing and Managing Behavior Models* for details.

A

Alltraps_logtofile behavior model 17
application status behavior model 97
Authentication behavior model 15

B

behavior models
 authentication failures 15
 Cisco Catalyst alarm 62
 Cisco Catalyst fan status 63
 Cisco Catalyst module status 64
 Cisco Catalyst power supplies 65
 Cisco configuration change 66
 Cisco CPU utilization 67
 Cisco fan status 68
 Cisco memory pool status 69
 Cisco power supply status 70
 Cisco router buffer 73
 Cisco router CPU 75
 Cisco router CRC 77
 Cisco router fan 79
 Cisco router memory 80
 Cisco router packets 82
 Cisco router power supply 84
 Cisco router temperature 86
 Cisco temperature status 71
 Cisco voltage status 72
 Compaq logical drive 88
 Compaq network interface card (NIC) 90
 Compaq physical drive 91
 Compaq SCSI drive 92
 Compaq uninterruptible power supply (UPS) 94
 Compaq unknown traps 94
 customizing 11
 discovery 17
 discovery of nodes (NT) 23
 disk usage 29

gateways 19
Inform Acknowledgement Queue
 connection 57
interface attribute values 20
IP sweep (NT) 23
IPSweep 17
log ifEntry to database 27
memory usage 30
node routing 26
OVPA connection 57
ping sweep (NT) 23
processor load 31
routing monitoring 26
SNMP Research application status 97
SNMP Research failed login events 103
SNMP Research file system status 99
SNMP Research log file events 101
SNMP Research memory status 105
SNMP Research processor load 106
Wellfleet (Nortel) router buffer 108
Wellfleet (Nortel) router fan 110
Wellfleet (Nortel) router memory 111
Wellfleet (Nortel) router power supply 113
Wellfleet (Nortel) router temperature 115
buffer (router) behavior models 73, 108

C

Catalyst behavior models 62

Catalyst power supply behavior models 65
CatalystAlarm behavior model 62
CatalystFanStatus behavior model 63
CatalystModuleStatus behavior model 64
CatalystPs1Status behavior model 65
CatalystPs2Status behavior model 65
CatalystPsTestResult behavior model 65
CIAgent behavior models 97
Cisco environmental models 66
Cisco router behavior models 73, 75
CiscoBufferUtilization behavior model 73
CiscoConfigChange behavior model 66
CiscoCPUUtilization behavior model 75
CiscoCpuUtilization behavior model 67
CiscoCRCCheck behavior model 77
CiscoFanCheck behavior model 79
CiscoFanStatus behavior model 68
CiscoMemGetCheck behavior model 80
CiscoMemoryPoolStatus behavior model 69
CiscoPacketsDrop behavior model 82
CiscoPwrSupplyCheck behavior model 84
CiscoPwrSupplyStatus behavior model 70
CiscoTempCheck behavior model 86
CiscoTempStatus behavior model 71
CiscoVoltageStatus behavior model 72
Compaq behavior models 88
CpqDaLogDrvStat behavior model 88

- CpqDaPhyDrvThresh behavior model 91
 - CpqNic behavior model 90
 - CpqScsiPhyStat behavior model 92
 - CpqUnknown behavior model 94
 - CpqUps behavior model 94
 - CPU (router) behavior models 75
 - CRC (router) behavior models 77
- D**
- data collection behavior models 17
 - dbi behavior model 27
 - DBI models 27
 - discovery behavior model 17, 23
 - see IPSweep 17
 - DiskUsage behavior model 29
 - documentation
 - conventions 5
 - feedback 6
 - drives
 - IDA drive (Compaq) behavior model 91
 - logical drive (Compaq) behavior model 88
 - physical drive (Compaq) behavior model 91
 - SCSI drive (Compaq) behavior model 92
- F**
- failed login events behavior model 103
 - fan (router) behavior models 79, 110
 - file system status behavior model 99
- G**
- gateway behavior model 19
 - GatewayList behavior model 19
- H**
- Host Resource models 29
- I**
- IDA drive (Compaq) behavior model 91
 - IfData_LogToFile behavior model 20
 - Ifdata_logtofile behavior model 17
 - ifEntry_LogToDBI behavior model 27
 - Inform Acknowledgement Queue behavior model 57
 - InformConnectStatus behavior model 57
 - interface attribute values behavior model 20
 - Interface status behavior models 33
 - IPSweep 17
 - IPSweep (NT) behavior model 23
 - IPSWEEP.EXE 23
- L**
- load, processor behavior model 106
 - log file events behavior model 101
 - logical drive behavior model 88
- M**
- memory (router) behavior models 80, 111
 - memory status behavior model 105
 - MemUsage behavior model 30
- N**
- network authentication failures behavior model 15
 - network interface card (NIC) behavior model 90
 - node routing behavior model 26
 - Node status behavior models 47
 - Nortel *See* Wellfleet router behavior models
- O**
- online knowledgebase 7
 - OVPA connection behavior model 57
- P**
- packets (router) behavior models 82
 - physical drive behavior model 91
 - ping sweep behavior model 23
 - power supply (Compaq) behavior models 94
 - power supply (router) behavior models 84
 - power supply behavior models 113
 - processor load behavior model 106
 - ProcessorLoad behavior model 31
- R**
- routers
 - Cisco buffer behavior model 73
 - Cisco CPU behavior model 75
 - Cisco fan behavior model 79
 - Cisco memory behavior model 80
 - Cisco packets behavior model 82
 - Cisco power supply behavior model 84
 - Cisco temperature behavior model 86
 - CiscoCRC behavior model 77
 - Wellfleet (Nortel) buffer behavior model 108
 - Wellfleet (Nortel) fan behavior model 110
 - Wellfleet (Nortel) memory behavior model 111
 - Wellfleet (Nortel) power supply behavior model 113
 - Wellfleet (Nortel) temperature behavior model 115
 - RoutingMon behavior model 26
- S**
- SCSI drive behavior model 92
 - SNMP Research behavior models 97
 - SNMP v3 behavior models 97
 - SnmprcsCritAppStatus behavior model 97
 - SnmprcsDiskUsage behavior model 99
 - SnmprcsGenericLogMonitor behavior model 101
 - SnmprcsLoginMonitor behavior model 103
 - SnmprcsMemUsage behavior model 105
 - SnmprcsProcessorLoad behavior model 106
 - status, application behavior model 97
 - status, file system behavior model 99

status, memory behavior model 105

T

TCP status behavior models 51

technical support 6

 contacting 7

 educational services 7

 professional services 6

temperature (router) behavior

 models 86, 115

traps

 behavior model for unknown traps
 (Compaq) 94

troubleshooting behavior models 57

U

uninterruptible power supply (UPS)

 behavior model 94

Up/Down interface status behavior
model 40

Up/Down Interface Status by Type
behavior model 42

W

Wellfleet (Nortel) behavior

 models 108

WfBufferUtilization behavior

 model 108

WfFanCheck behavior model 110

WfMemCheck behavior model 111

WfPwrSupplyCheck behavior

 model 113

WfTempCheck behavior model 115

