



# HP Network Node Manager i Software 10.00

## Step-by-Step Guide to Pairwise and Batch Incident Configuration

### Table of Contents

Introduction.....2

Pairwise .....2

    Pairwise Example 1 (Same Trap with Different Varbinds).....2

        Initial Preparation .....2

        Load the MIBs and the Traps .....2

        Configure Trap Format .....4

        Manually Sending the Traps.....6

        Same Trap, Different Varbinds Pairwise Configuration.....7

        Testing..... 10

    Pairwise Example 2 (Three Traps) ..... 13

Batch Incident Configuration ..... 17

## Introduction

This whitepaper uses examples to document pairwise functionality and batch incident configuration tools available in NNMi 10.00.

## Pairwise

This section includes two examples that document pairwise functionality. The first example involves trap pairing using the same trap yet having different varbinds to differentiate the meaning of the trap. The second example describes a trap pairing with two other traps. (Prior to NNMi 9.20, a trap could only be involved in a single pairing.) This example includes the use of the “time duration” feature and the “delete when canceled” feature.

### Pairwise Example 1 (Same Trap with Different Varbinds)

#### Initial Preparation

Set up the NNMi management server to receive the traps of interest. This example uses traps from Ascend Communications. This was selected because it is a MIB not shipped with NNMi, which allows this example to illustrate many preparation steps. The sequence of traps and varbinds shown may not represent an actual sequence. All traps are sent using NNMi command line scripts rather than an actual network device.

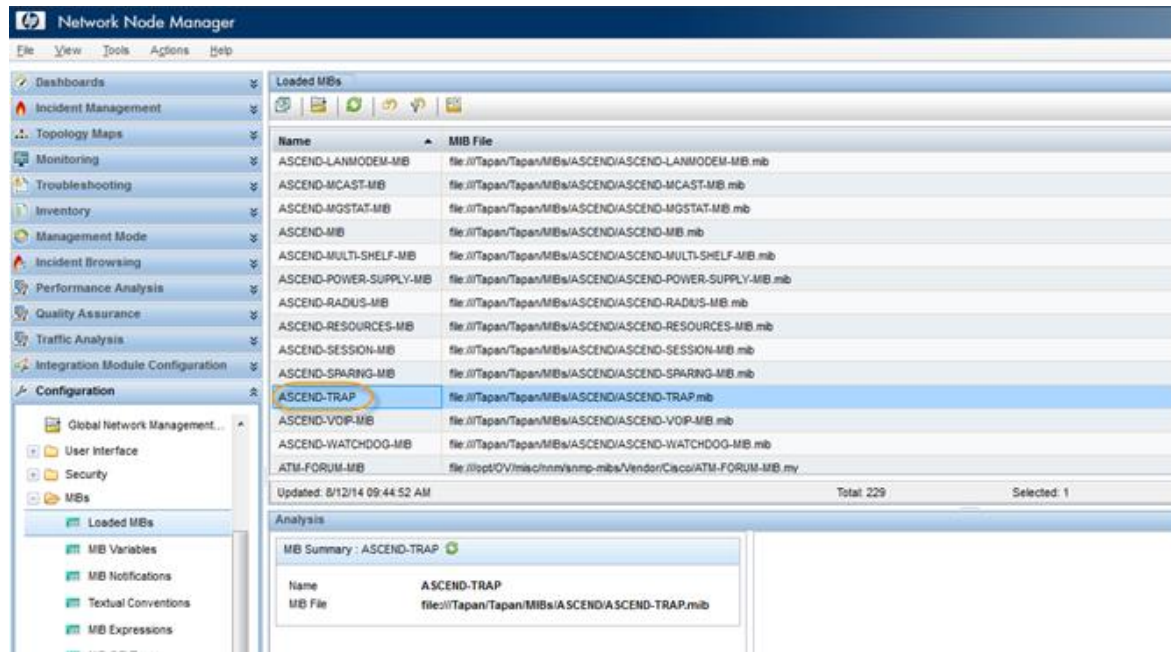
#### Load the MIBs and the Traps

1. Load the dependent MIBs using the `nnmloadmib.ovpl` command. The dependent MIBs include the following:
  - IF-MIB.mib
  - RFC1213-MIB.mib
  - ASCEND-MIB.mib
  - ASCEND-CHASSIS-MIB.mib
  - ASCEND-RADIUS-MIB.mib
  - ASCEND-MCAST-MIB.mib
  - ASCEND-LANMODEM-MIB.mib
  - ASCEND-SESSION-MIB.mib
  - ASCEND-POWER-SUPPLY-MIB.mib
  - ASCEND-MULTI-SHELF-MIB.mib
  - ASCEND-ATMP-MIB.mib
  - ASCEND-RESOURCES-MIB.mib
  - ASCEND-WATCHDOG-MIB.mib
  - ASCEND-CALL-LOGGING-MIB.mib
  - ASCEND-VOIP-MIB.mib
  - ASCEND-CALL-LOGGING-MIB.mib
  - ASCEND-ADVANCED-AGENT-MIB.mib
  - RFC-1215.mib
  - ASCEND-MGSTAT-MIB.mib
  - ASCEND-SPARING-MIB.mib
2. After loading the dependent MIBs, load the desired MIB (`ASCEND-TRAP.mib`) as shown.
 

```
# nnmloadmib.ovpl -load ASCEND-TRAP.mib Successfully completed operation
LoadMib.

45 Traps were loaded.
```
3. The output indicates that 45 traps were loaded; however, they are not completely ready for use. You must first obtain the name of the MIB module using the NNMi console. Navigate to the **Configuration** workspace and select **Loaded MIBs**. Then locate the module name associated with the MIB file just loaded. You can see in the following figure that the module name is ASCEND-TRAP.

## Loaded MIBs



- Load the traps using the `nnmincidentcfg.ovpl` command along with the module name as shown in the following:

```
# nnmincidentcfg.ovpl -loadTraps ASCEND-TRAP
```

SNMP trap(s) from mib module loaded: ASCEND-TRAP.  
Number of traps: 45.

The following traps were added to incident configuration:

```
wanLineStateChange - .1.3.6.1.4.1.529.0.40
portConnected - .1.3.6.1.4.1.529.0.7
callLogDroppedPkt - .1.3.6.1.4.1.529.0.41
multicastHeartBeatMonitor - .1.3.6.1.4.1.529.0.19
maxTelnetAttempts - .1.3.6.1.4.1.529.0.15
powerSupplyStateChange - .1.3.6.1.4.1.529.0.23
atmpAgentErrorRecvTrap - .1.3.6.1.4.1.529.0.29
portWaiting - .1.3.6.1.4.1.529.0.6
radiusServerChange - .1.3.6.1.4.1.529.0.18
powerSupplyOperationalStateChange - .1.3.6.1.4.1.529.0.24
consoleStateChange - .1.3.6.1.4.1.529.0.12
dirdoListFailure - .1.3.6.1.4.1.529.0.21
portAcrPending - .1.3.6.1.4.1.529.0.10
lanModemMovedToSuspectList - .1.3.6.1.4.1.529.0.20
atmpMaxTunnelExceeded - .1.3.6.1.4.1.529.0.27
sysSlotStateChange - .1.3.6.1.4.1.529.0.22
suspectAccessResource - .1.3.6.1.4.1.529.0.34
portWaitSerial - .1.3.6.1.4.1.529.0.2
portCollectDigits - .1.3.6.1.4.1.529.0.5
sdtnPrimaryListEmptyTrap - .1.3.6.1.4.1.529.0.31
megacoLinkStatusTrap - .1.3.6.1.4.1.529.0.42
portCarrier - .1.3.6.1.4.1.529.0.8
portUseExceeded - .1.3.6.1.4.1.529.0.13
```

```

sparingIfStatusChange - .1.3.6.1.4.1.529.0.44
portRinging - .1.3.6.1.4.1.529.0.4
portLoopback - .1.3.6.1.4.1.529.0.9

controllerSwitchoverTrap - .1.3.6.1.4.1.529.0.37
voipGkChange - .1.3.6.1.4.1.529.0.39

portDteNotReady - .1.3.6.1.4.1.529.0.11
watchdogWarningTrap - .1.3.6.1.4.1.529.0.35

callLogServChange - .1.3.6.1.4.1.529.0.38
eventTableOverwrite - .1.3.6.1.4.1.529.0.16

portDualDelay - .1.3.6.1.4.1.529.0.1
systemUseExceeded - .1.3.6.1.4.1.529.0.14

sysConfigChangeTrap - .1.3.6.1.4.1.529.0.30
portInactive - .1.3.6.1.4.1.529.0.0

cntrReduAvailTrap - .1.3.6.1.4.1.529.0.45
atmpAgentErrorSentTrap - .1.3.6.1.4.1.529.0.28

sysLastRestartReasonTrap - .1.3.6.1.4.1.529.0.26
sparingSlotStatusChange - .1.3.6.1.4.1.529.0.43

slotCardResetTrap - .1.3.6.1.4.1.529.0.36
systemClockDrifted - .1.3.6.1.4.1.529.0.33

sdtnSecondaryListEmptyTrap - .1.3.6.1.4.1.529.0.32
multiShelfStateChange - .1.3.6.1.4.1.529.0.25

portHaveSerial - .1.3.6.1.4.1.529.0.3

```

The traps are now completely loaded into NNMi.

## Configure Trap Format

In this section, you can see that the different varbind values differentiate the pairwise behavior.

1. Configure the format of the sysSlotStateChange trap (the example trap) to better see the varbind values. In the **Configuration** workspace, expand the **Incidents** folder, select **SNMP Trap Configurations**, and double-click **sysSlotStateChange**.

### SNMP Trap Configurations

Name	SNMP Object ID	Enabled	Root Cause	Deduplicat Enabled	Rate Enabled	Seve	Cate	Fami	Author	Message Format
STPNewRoot	.1.3.6.1.2.1.17.0.1	-	-	✓	-	▲	V	F	HP Network Nod	STP New Root
STPTopologyChange	.1.3.6.1.2.1.17.0.2	-	-	✓	-	▲	V	F	HP Network Nod	STP Topology Chan
suspectAccessResource	.1.3.6.1.4.1.529.0.34	✓	-	-	-	●	N	7	Customer	suspectAccessRet
sysConfigChangeTrap	.1.3.6.1.4.1.529.0.30	✓	-	-	-	●	N	7	Customer	sysConfigChangeT
sysLastRestartReasonTrap	.1.3.6.1.4.1.529.0.26	✓	-	-	-	●	N	7	Customer	sysLastRestartRea
SyslogMessage	.1.3.6.1.4.1.9.9.41.2.0.1	-	-	-	-	▲	V	F	HP Network Nod	\$1:\$3 \$4 (syslog)
sysSlotStateChange	.1.3.6.1.4.1.529.0.22	✓	-	-	-	●	N	7	Customer	sysSlotStateChange
systemClockDrifted	.1.3.6.1.4.1.529.0.33	✓	-	-	-	●	N	7	Customer	systemClockDrifted
systemUseExceeded	.1.3.6.1.4.1.529.0.14	✓	-	-	-	●	N	7	Customer	systemUseExceeds
voipGkChange	.1.3.6.1.4.1.529.0.39	✓	-	-	-	●	N	7	Customer	voipGkChange

Updated: 10/17/11 12:53:14 PM MDT Total: 112 Selected: 1

**SNMP Trap Configuration Summary - sysSlotStateChange**

SNMP Object ID: .1.3.6.1.4.1.529.0.22

Message Format: sysSlotStateChange

**Details**

Enabled: true

Severity: Normal

Status: Status


Category: None

Family: Customer

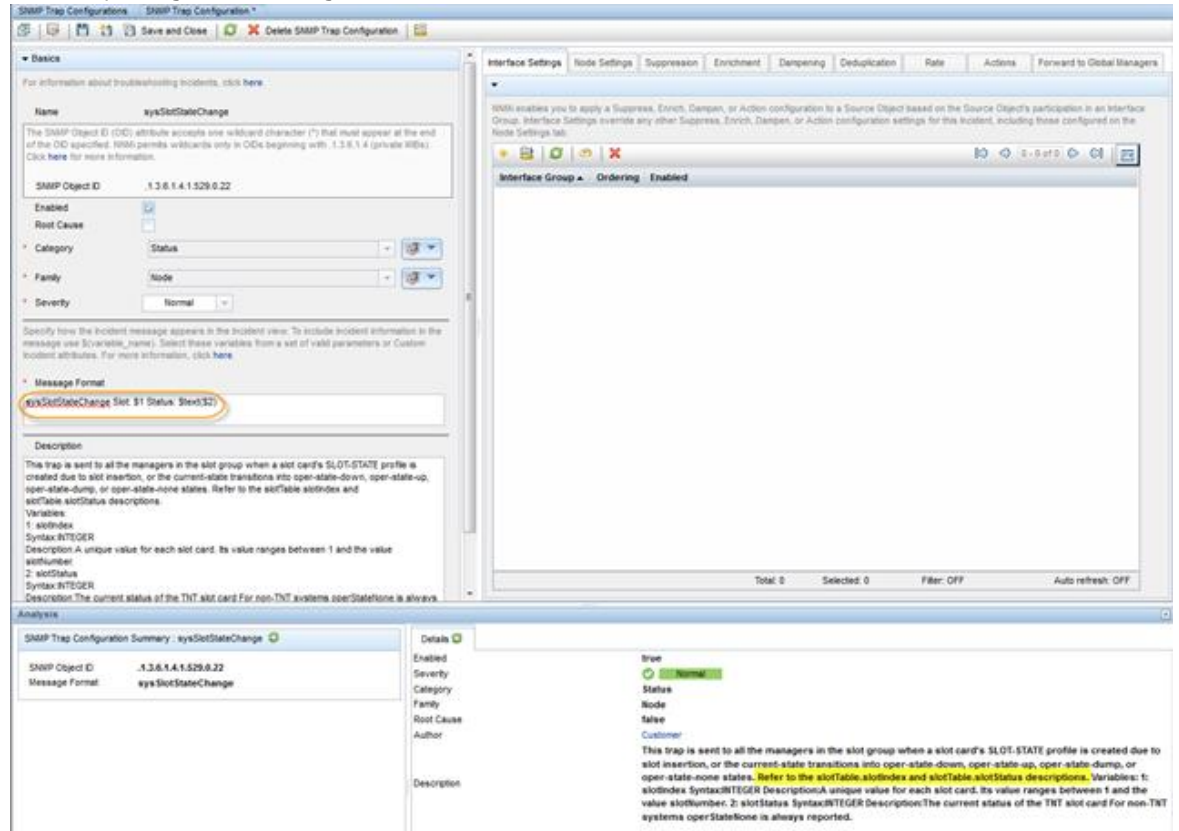
Root Cause: false

Author: Customer

Description: This trap is sent to all the manu slot insertion, or the current-st

- Change the format to include the slot number and the current state as shown in the following figure. Notice in the Message Format area the following text: "Slot: \$1 Status: \$ text (\$2)". The first part, "\$1", represents the first varbind. The second part, "\$ text (\$2)", tells NNMi to print the textual representation of the second varbind rather than the number. This makes the traps more readable in the NNMi console. Click  **Save and Close**.

### SNMP Trap Configuration: Message Format



**SNMP Trap Configuration: Message Format**

**Basics**

For information about troubleshooting incidents, click [here](#).

**Name:** sysSlotStateChange

The SNMP Object ID (OID) attribute accepts one wildcard character (\*) that must appear at the end of the OID specified. NNMi permits wildcards only in OIDs beginning with 1.3.6.1.4 (private MIBs). Click [here](#) for more information.

**SNMP Object ID:** 1.3.6.1.4.1.329.0.22

**Enabled:** ☒

**Root Cause:** ☐

**Category:** Status

**Family:** Node

**Severity:** Normal

Specify how the incident message appears in the incident view. To include incident information in the message use {variable\_name}. Select those variables from a set of valid parameters or Custom Incident attributes. For more information, click [here](#).

**Message Format:** sysSlotStateChange Slot: \$1 Status: \$text(\$2)

**Description:**

This trap is sent to all the managers in the slot group when a slot card's SLOT-STATE profile is created due to slot insertion, or the current-state transitions into open-state-down, open-state-up, open-state-dump, or open-state-none states. Refer to the slotTable.slotIndex and slotTable.slotStatus descriptions.

**Variables:**

1: slotIndex  
Syntax: INTEGER  
Description: A unique value for each slot card. Its value ranges between 1 and the value slotNumber.

2: slotStatus  
Syntax: INTEGER  
Description: The current status of the T1T slot card. For non-T1T systems openStateNone is always reported.

**Analysis**

**SNMP Trap Configuration Summary: sysSlotStateChange**

SNMP Object ID: 1.3.6.1.4.1.329.0.22  
Message Format: sysSlotStateChange

**Details**

Enabled: true  
Severity: Normal  
Category: Status  
Family: Node  
Root Cause: false  
Author: Customer

**Description:**

This trap is sent to all the managers in the slot group when a slot card's SLOT-STATE profile is created due to slot insertion, or the current-state transitions into open-state-down, open-state-up, open-state-dump, or open-state-none states. Refer to the slotTable.slotIndex and slotTable.slotStatus descriptions. Variables: 1: slotIndex Syntax: INTEGER Description: A unique value for each slot card. Its value ranges between 1 and the value slotNumber. 2: slotStatus Syntax: INTEGER Description: The current status of the T1T slot card. For non-T1T systems openStateNone is always reported.

- To see the values of slot status, you can look up slotStatus in the MIB variables. Go to **Configuration -> MIBs -> MIB Variables** and search for slotStatus. In the Analysis pane shown in figure below, you can see that the enumerated values include the following:  

```
operStateDown = 1,
operStateUp = 2,
operStateDiag = 3,
operStateCoreDump = 4,
operStateLoading = 5,
operStatePost = 6,
operStateNone = 7,
operStateOccupied = 8
```

## MIB Variables

The screenshot displays the NNMi MIB Variables window. The left sidebar shows the 'Configuration' menu with 'MIBs' selected. The main pane shows a table of MIB variables. The variable 'slotStatus' is selected, and its details are shown in the right pane. The details pane includes the variable's OID, syntax, and a list of enumerated values (operStateDown, operStateUp, operStateDiag, operStateCoreDump, operStateLoading, operStatePost, operStateNone, operStateOccupied) with their corresponding values (1-8). A note indicates that for non-TWT systems, operStateNone is always reported.

Now NNMi is configured to be able to receive the traps and easily read them in the NNMi console.

### Manually Sending the Traps

For this example, there is not a device sending the traps, so you must manually create and send them.

**Tip:** It is always best to use traps sent directly from a device in the network but manually sending them is a good way to develop and test your solution.

For this example, use the `nnmsnmpnotify.ovpl` command to format the trap that is sent. See the following format of the command used to send the `operStateUp` for slot 1 (note that this command must all be on one line rather than the three lines shown in the following sample).

**Tip:** See the `nnmsnmpnotify.ovpl` reference page or the UNIX manpage for more information.

Make sure your source node (in this case, `ciscope6524`) is already discovered in NNMi to receive the trap.

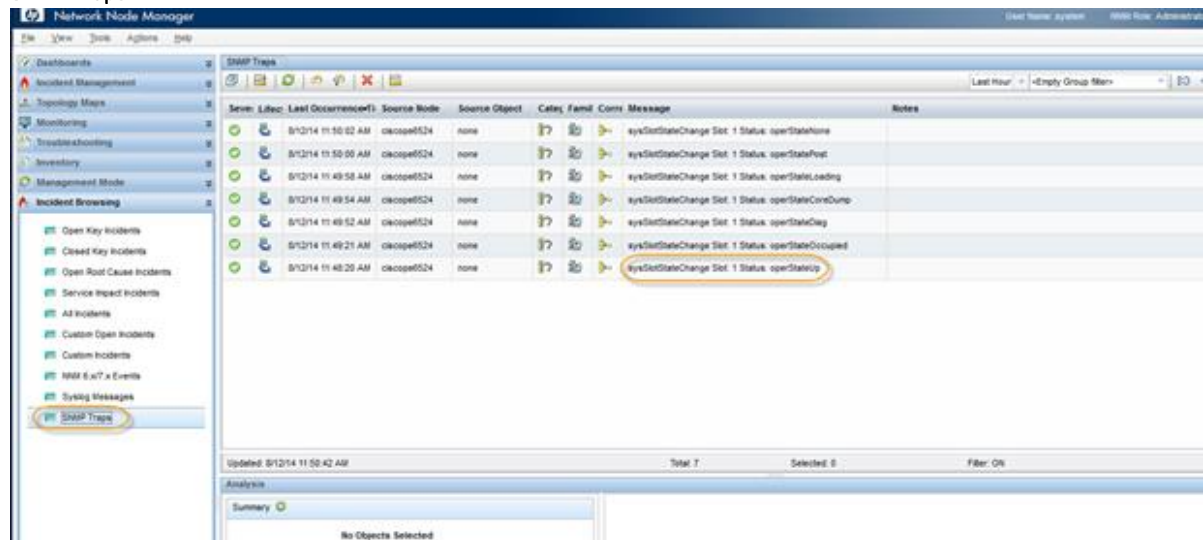
```
nnmsnmpnotify.ovpl -v 1 -a ciscope6524 localhost .1.3.6.1.4.1.529.0.22
.1.3.6.1.4.1.529.2.2.1.1 integer 1
.1.3.6.1.4.1.529.2.2.1.8 integer 2
```

You can send status messages to ensure traps are coming in successfully by changing the second varbind value to one represents a different state. For example, you can change the value to 8, which represents `operStateOccupied`.

```
nnmsnmpnotify.ovpl -v 1 -a ciscope6524 localhost .1.3.6.1.4.1.529.0.22
.1.3.6.1.4.1.529.2.2.1.1 integer 1
.1.3.6.1.4.1.529.2.2.1.8 integer 8
```

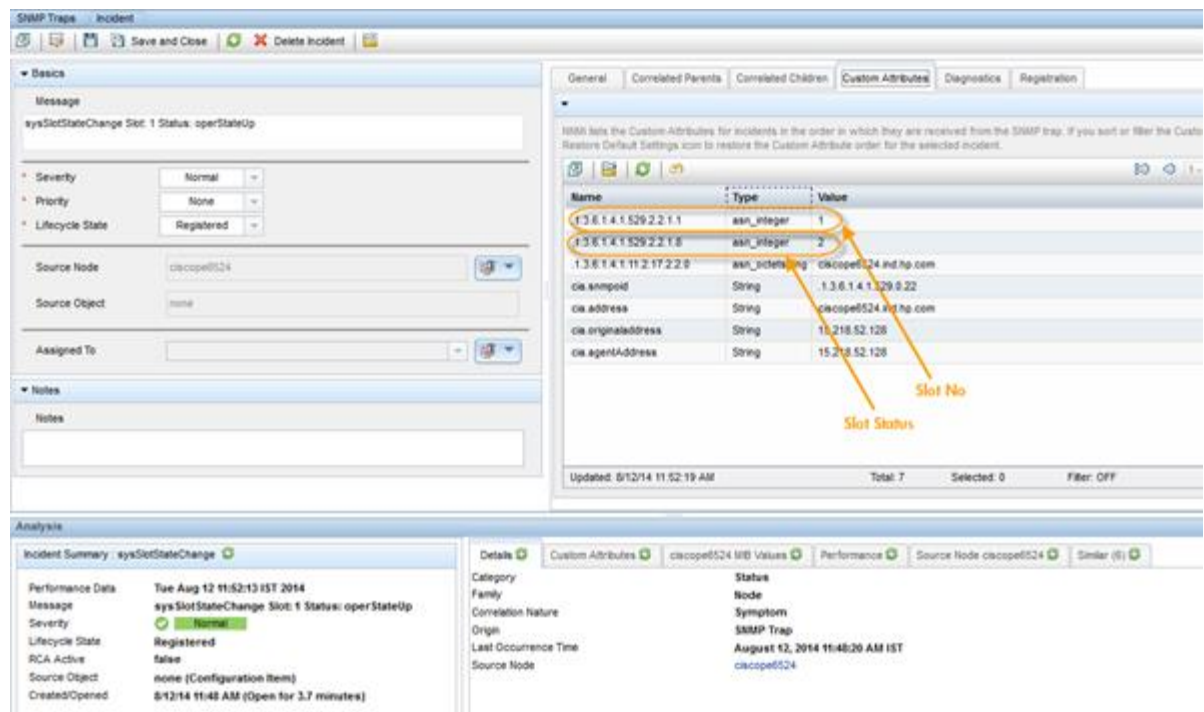
As indicated in the figure below, traps containing all of the supported values 1-8 have been sent. To see these traps, go to **Incident Browsing -> SNMP Traps**. When looking for these traps in the SNMP Traps table, make sure that you have selected an appropriate time filter to include the traps you want to see.

## SNMP Traps



If you open (double-click) one of these traps, you can see the varbinds associated with the trap. See the following figure.

## Custom Attributes



## Same Trap, Different Varbinds Pairwise Configuration

Suppose that when a status of operStateUp arrives, you want to cancel any of the other state traps for this node. The correlation searches back in time for up to 24 hours for any incidents to cancel. For example, an operStateUp value cancels any traps that have any of the other statuses for the same source node and slot number for the past 24 hours (this is an example scenario only). You can also require that the Source Node and the Slot Number be the same on both traps.

1. Go to **Incidents -> Pairwise Configurations** and click the  icon.



## Pairwise Configurations

Name	Enabled	First Incident	Second Incident
CiscoLinkDownUpPair	✓	CiscoLinkDown	CiscoLinkUp
CiscoModuleDownUpPair	✓	CiscoModuleDown	CiscoModuleUp
DEVI4/FAN_FAILED_RECOVERED-P	✓	DEVI4/FAN_FAILED	DEVI4/FAN_RECOVERED
DEVI4/POWER_FAILED_RECOVERED	✓	DEVI4/POWER_FAILED	DEVI4/POWER_RECOVERED
Lineproto-5-UpDownPair	✓	LINEPROTO-5-UPDOWN	LINEPROTO-5-UPDOWN
Link-3-UpDownPair	✓	LINK-3-UPDOWN	LINK-3-UPDOWN
OPTMOD/4/MODULE_OUT_N-Pair	✓	OPTMOD/4/MODULE_OUT	OPTMOD/4/MODULE_IN
OPTMOD/5/MOD_ALM_ON_OFF-Pair	✓	OPTMOD/5/MOD_ALM_ON	OPTMOD/5/MOD_ALM_OFF
ProCurve-RMON_LACP_DYNAMIC_T	✓	ProCurve-RMON_LACP_D	ProCurve-RMON_LACP_DYNAMIC_TRUNK_ON_LINE
ProCurve-RMON_SSH_DISABLED_E	✓	ProCurve-RMON_SSH_DISABLED	ProCurve-RMON_SSH_ENABLED
QaSpSite	✓	SiteDown	SiteUp
QaSpSiteReachable	✓	SiteNotReachable	SiteReachable
QaSpSiteServiceUp	✓	ServiceToSiteDown	ServiceToSiteUp

Updated: 8/12/14 11:55:34 AM      Total: 23      Selected: 0

Analysis  
Summary

- Set the First Incident and Second Incident to be sysSlotStateChange. In this case, these are the same trap. Only the varbind value differentiates them.
- Set the duration time to 24 hours.
- Because this is not just a simple pair of traps, it might be best to start with defining the Second Incident Payload Filter. The second incident is the trap that closes the other traps on the back end.
- Build an AND condition using ciaName and ciaValue. (Custom Incident Attributes (CIAs) are varbinds in NNMi). Create the condition where the varbind Object ID (OID) equals .1.3.6.1.4.1.529.2.2.1.8 and the varbind value equals 2. Remember that the value of 2 represents operStateUp.



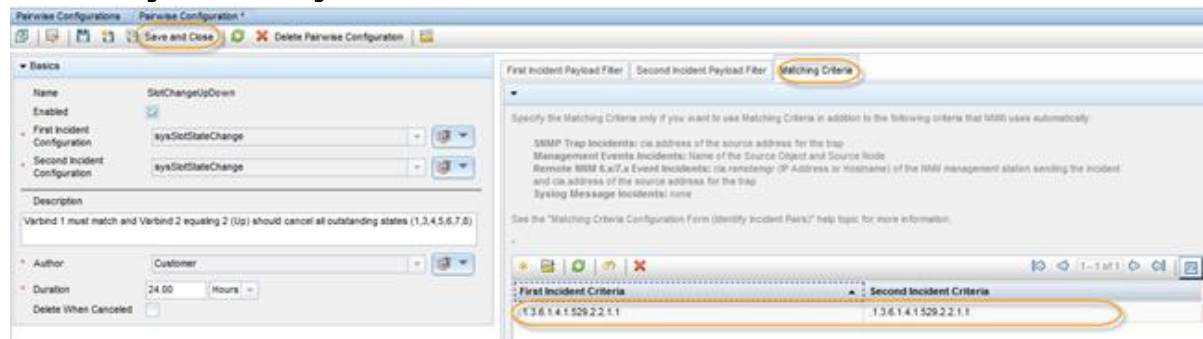
## Pairwise Configuration: Filter Editor

- Next, define the First Incident Payload Filter. This represents any traps that are to be closed as part of this pairing. Be careful entering values with the “in” operator (do not use commas; instead, put each value on a separate line). The AND condition shown in figure below, represents any of the traps with a varbind value of operStateDown, operStateDiag, operStateCoreDump, operStateLoading, operStatePost, operStateNone, or operStateOccupied (1, 3, 4, 5, 6, 7 or 8).

## Pairwise Configuration: First Incident Payload Filter

- Finally, enter the Matching Criteria. NNMi automatically performs a source node match for all pairwise operations, so we do not need to specify the source node as part of the matching criteria. We enter the OID for the slot number since that must be the same for the match to be valid. Click **Save and Close** to finish this pairwise configuration.

## Pairwise Configuration: Matching Criteria



### Testing

1. Test the pairwise configuration by manually sending the traps. All of the traps are sent except the one with a value of 2 (operStateUp).

```
# nnmsnmnotify.ovpl -v 1 -a ciscope6524.ind.hp.com localhost
.1.3.6.1.4.1.529.0.22 .1.3.6.1.4.1.529.2.2.1.1 integer 1
.1.3.6.1.4.1.529.2.2.1.8 integer 1

# nnmsnmnotify.ovpl -v 1 -a ciscope6524.ind.hp.com localhost
.1.3.6.1.4.1.529.0.22 .1.3.6.1.4.1.529.2.2.1.1 integer 1
.1.3.6.1.4.1.529.2.2.1.8 integer 3

# nnmsnmnotify.ovpl -v 1 -a ciscope6524.ind.hp.com localhost
.1.3.6.1.4.1.529.0.22 .1.3.6.1.4.1.529.2.2.1.1 integer 1
.1.3.6.1.4.1.529.2.2.1.8 integer 4

# nnmsnmnotify.ovpl -v 1 -a ciscope6524.ind.hp.com localhost
.1.3.6.1.4.1.529.0.22 .1.3.6.1.4.1.529.2.2.1.1 integer 1
.1.3.6.1.4.1.529.2.2.1.8 integer 5

# nnmsnmnotify.ovpl -v 1 -a ciscope6524.ind.hp.com localhost
.1.3.6.1.4.1.529.0.22 .1.3.6.1.4.1.529.2.2.1.1 integer 1
.1.3.6.1.4.1.529.2.2.1.8 integer 6

# nnmsnmnotify.ovpl -v 1 -a ciscope6524.ind.hp.com localhost
.1.3.6.1.4.1.529.0.22 .1.3.6.1.4.1.529.2.2.1.1 integer 1
.1.3.6.1.4.1.529.2.2.1.8 integer 7

# nnmsnmnotify.ovpl -v 1 -a ciscope6524.ind.hp.com localhost
.1.3.6.1.4.1.529.0.22 .1.3.6.1.4.1.529.2.2.1.1 integer 1
.1.3.6.1.4.1.529.2.2.1.8 integer 8
```

2. Look in the trap table to see the traps.

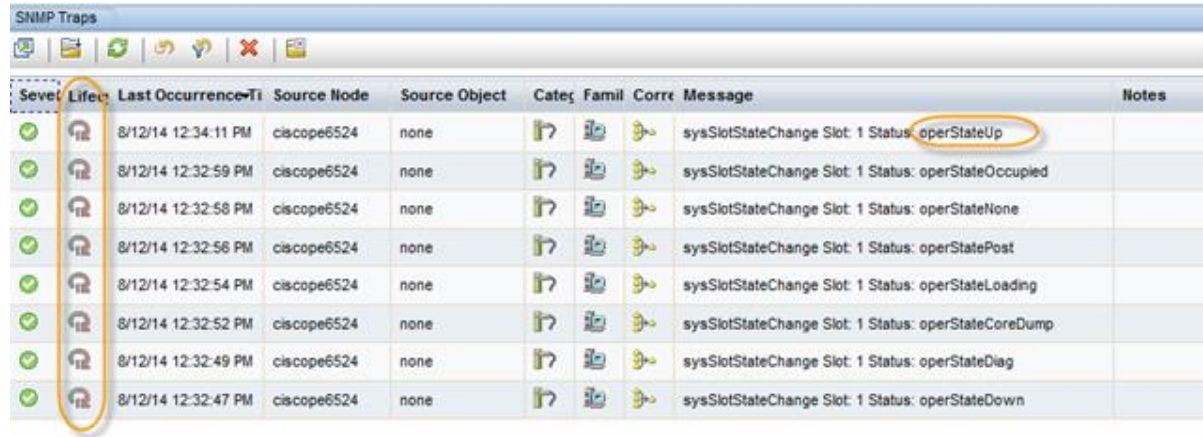
### SNMP Traps

Severity	LifeCycle	Last Occurrence	Time	Source Node	Source Object	Catag	Famil	Corr	Message	Notes
✓	🔄	8/12/14 12:32:59 PM		ciscope6524	none	🔄	🔄	🔄	sysSlotStateChange Slot: 1	Status: operStateOccupied
✓	🔄	8/12/14 12:32:58 PM		ciscope6524	none	🔄	🔄	🔄	sysSlotStateChange Slot: 1	Status: operStateNone
✓	🔄	8/12/14 12:32:56 PM		ciscope6524	none	🔄	🔄	🔄	sysSlotStateChange Slot: 1	Status: operStatePost
✓	🔄	8/12/14 12:32:54 PM		ciscope6524	none	🔄	🔄	🔄	sysSlotStateChange Slot: 1	Status: operStateLoading
✓	🔄	8/12/14 12:32:52 PM		ciscope6524	none	🔄	🔄	🔄	sysSlotStateChange Slot: 1	Status: operStateCoreDump
✓	🔄	8/12/14 12:32:49 PM		ciscope6524	none	🔄	🔄	🔄	sysSlotStateChange Slot: 1	Status: operStateDiag
✓	🔄	8/12/14 12:32:47 PM		ciscope6524	none	🔄	🔄	🔄	sysSlotStateChange Slot: 1	Status: operStateDown

3. Send the "up" trap and it should close all the other traps that happened within the last 24 hours. Note that you may not see this happen immediately in the NNMi console. The pairwise thread runs every 30 seconds (or when a certain count is reached, if that happens sooner than 30 seconds). So you may not see the correlations for 30 seconds after sending

the trap. Note that if you are using northbound integration with NNMi, this pairwise feature will not keep the correlated traps from “going north”. You can prevent them from going north by using the dampening feature (see the *NNMi Deployment Reference* for more information on dampening).

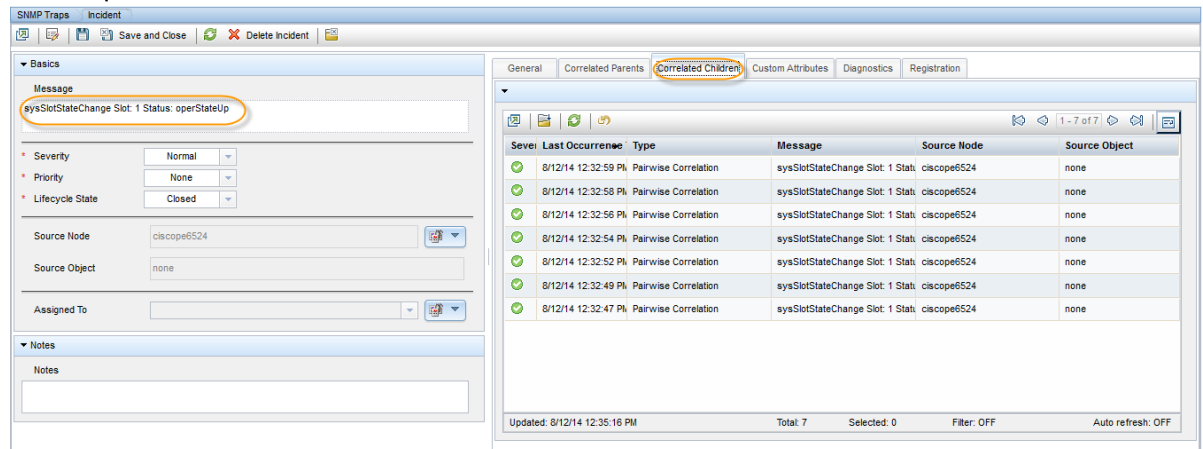
## SNMP Traps



Severity	Lifecycle	Last Occurrence	Time	Source Node	Source Object	Category	Family	Correlation	Message	Notes
✓	Up	8/12/14 12:34:11 PM		ciscope6524	none	7	1	3	sysSlotStateChange Slot: 1 Status: operStateUp	
✓	Up	8/12/14 12:32:59 PM		ciscope6524	none	7	1	3	sysSlotStateChange Slot: 1 Status: operStateOccupied	
✓	Up	8/12/14 12:32:58 PM		ciscope6524	none	7	1	3	sysSlotStateChange Slot: 1 Status: operStateNone	
✓	Up	8/12/14 12:32:56 PM		ciscope6524	none	7	1	3	sysSlotStateChange Slot: 1 Status: operStatePost	
✓	Up	8/12/14 12:32:54 PM		ciscope6524	none	7	1	3	sysSlotStateChange Slot: 1 Status: operStateLoading	
✓	Up	8/12/14 12:32:52 PM		ciscope6524	none	7	1	3	sysSlotStateChange Slot: 1 Status: operStateCoreDump	
✓	Up	8/12/14 12:32:49 PM		ciscope6524	none	7	1	3	sysSlotStateChange Slot: 1 Status: operStateDiag	
✓	Up	8/12/14 12:32:47 PM		ciscope6524	none	7	1	3	sysSlotStateChange Slot: 1 Status: operStateDown	

- If you double-click the “up” trap and look at the **Correlated Children** tab, you can see all of the traps that were closed due to the pairwise correlation.

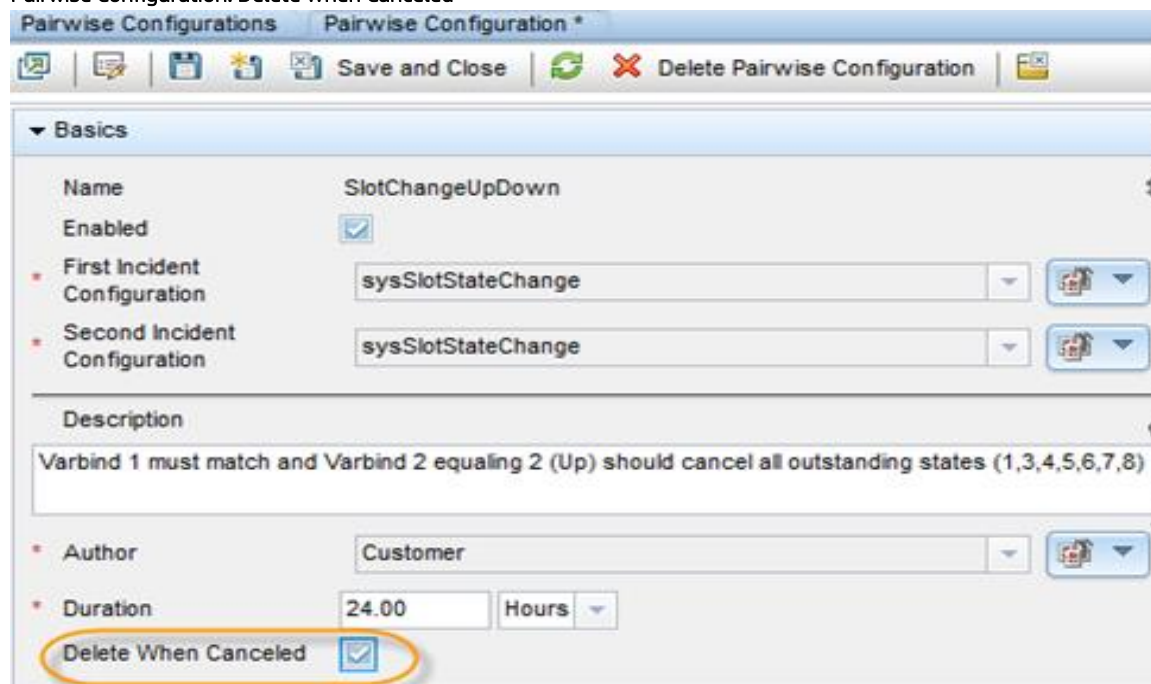
## SNMP Traps: Incident: Correlated Children



Severity	Last Occurrence	Type	Message	Source Node	Source Object
✓	8/12/14 12:32:59 PM	Pairwise Correlation	sysSlotStateChange Slot: 1 Status: operStateUp	ciscope6524	none
✓	8/12/14 12:32:58 PM	Pairwise Correlation	sysSlotStateChange Slot: 1 Status: operStateOccupied	ciscope6524	none
✓	8/12/14 12:32:56 PM	Pairwise Correlation	sysSlotStateChange Slot: 1 Status: operStateNone	ciscope6524	none
✓	8/12/14 12:32:54 PM	Pairwise Correlation	sysSlotStateChange Slot: 1 Status: operStatePost	ciscope6524	none
✓	8/12/14 12:32:52 PM	Pairwise Correlation	sysSlotStateChange Slot: 1 Status: operStateLoading	ciscope6524	none
✓	8/12/14 12:32:49 PM	Pairwise Correlation	sysSlotStateChange Slot: 1 Status: operStateCoreDump	ciscope6524	none
✓	8/12/14 12:32:47 PM	Pairwise Correlation	sysSlotStateChange Slot: 1 Status: operStateDiag	ciscope6524	none

**Tip:** You can configure NNMi to delete traps when they are canceled (closed). See the following figure.

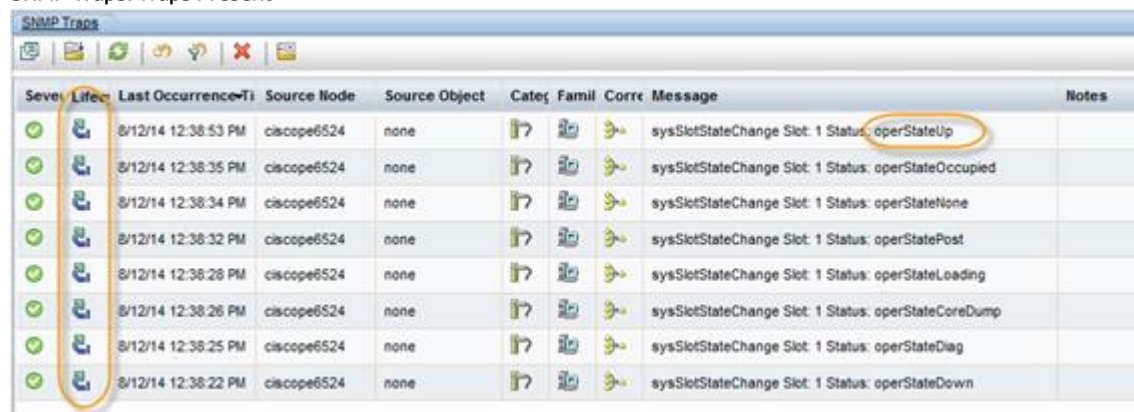
### Pairwise Configuration: Delete When Canceled



The screenshot shows the 'Pairwise Configuration' dialog box. The 'Basics' tab is selected. The 'Name' field is 'SlotChangeUpDown'. The 'Enabled' checkbox is checked. The 'First Incident Configuration' and 'Second Incident Configuration' are both set to 'sysSlotStateChange'. The 'Description' field contains the text: 'Varbind 1 must match and Varbind 2 equaling 2 (Up) should cancel all outstanding states (1,3,4,5,6,7,8)'. The 'Author' is set to 'Customer'. The 'Duration' is set to '24.00' hours. The 'Delete When Canceled' checkbox is checked and highlighted with an orange circle.

The deleting of events from the pairwise operation is done by another thread that runs every 2 minutes, so you might first see the traps in the browser.

### SNMP Traps: Traps Present



Severity	Life	Last Occurrence	Time	Source Node	Source Object	Categ	Famil	Corr	Message	Notes
✓	🔧	8/12/14 12:38:53 PM		ciscope6524	none	🔧	🔧	🔧	sysSlotStateChange Slot: 1 Status: operStateUp	
✓	🔧	8/12/14 12:38:35 PM		ciscope6524	none	🔧	🔧	🔧	sysSlotStateChange Slot: 1 Status: operStateOccupied	
✓	🔧	8/12/14 12:38:34 PM		ciscope6524	none	🔧	🔧	🔧	sysSlotStateChange Slot: 1 Status: operStateNone	
✓	🔧	8/12/14 12:38:32 PM		ciscope6524	none	🔧	🔧	🔧	sysSlotStateChange Slot: 1 Status: operStatePost	
✓	🔧	8/12/14 12:38:28 PM		ciscope6524	none	🔧	🔧	🔧	sysSlotStateChange Slot: 1 Status: operStateLoading	
✓	🔧	8/12/14 12:38:26 PM		ciscope6524	none	🔧	🔧	🔧	sysSlotStateChange Slot: 1 Status: operStateCoreDump	
✓	🔧	8/12/14 12:38:25 PM		ciscope6524	none	🔧	🔧	🔧	sysSlotStateChange Slot: 1 Status: operStateDiag	
✓	🔧	8/12/14 12:38:22 PM		ciscope6524	none	🔧	🔧	🔧	sysSlotStateChange Slot: 1 Status: operStateDown	

Then you might see them close.



## SNMP Traps: Traps Closed

Severity	LifeCycle	Last Occurrence	Time	Source Node	Source Object	Category	Family	Corrective	Message	Notes
✓	🔄	8/12/14 12:38:53 PM		ciscope6524	none	🔧	🔧	🔧	sysSlotStateChange Slot: 1 Status: operStateUp	
✓	🔄	8/12/14 12:38:35 PM		ciscope6524	none	🔧	🔧	🔧	sysSlotStateChange Slot: 1 Status: operStateOccupied	
✓	🔄	8/12/14 12:38:34 PM		ciscope6524	none	🔧	🔧	🔧	sysSlotStateChange Slot: 1 Status: operStateNone	
✓	🔄	8/12/14 12:38:32 PM		ciscope6524	none	🔧	🔧	🔧	sysSlotStateChange Slot: 1 Status: operStatePost	
✓	🔄	8/12/14 12:38:28 PM		ciscope6524	none	🔧	🔧	🔧	sysSlotStateChange Slot: 1 Status: operStateLoading	
✓	🔄	8/12/14 12:38:26 PM		ciscope6524	none	🔧	🔧	🔧	sysSlotStateChange Slot: 1 Status: operStateCoreDump	
✓	🔄	8/12/14 12:38:25 PM		ciscope6524	none	🔧	🔧	🔧	sysSlotStateChange Slot: 1 Status: operStateDiag	
✓	🔄	8/12/14 12:38:22 PM		ciscope6524	none	🔧	🔧	🔧	sysSlotStateChange Slot: 1 Status: operStateDown	

Then you will see them get deleted.

## SNMP Traps: Traps Deleted

Severity	LifeCycle	Last Occurrence	Time	Source Node	Source Object	Category	Family	Corrective	Message	Notes
----------	-----------	-----------------	------	-------------	---------------	----------	--------	------------	---------	-------

## Pairwise Example 2 (Three Traps)

You can construct sophisticated pairwise filters involving multiple traps. For example, you might have three different traps (X, Y, and Z). You could have Z close both X and Y. To do this with different traps, you would need to create separate pairwise configurations for each relationship: Z closing X, and Z closing Y. Suppose that when you receive a portCarrier trap, you want to cancel both the portConnected and portWaiting traps.

## SNMP Trap Configurations

Name	SNMP Object ID	Enabled	Root Cause	Deduplication Enabled	Rate Enabled	Severity	Category	Family	Author	Message Format
maxTelnetAttempts	.1.3.6.1.4.1.529.0.15	✓	-	-	-	🟢	🔧	🔧	Customer	maxTelnetAttempts
megacoLinkStatusTrap	.1.3.6.1.4.1.529.0.42	✓	-	-	-	🟢	🔧	🔧	Customer	megacoLinkStatusTrap
multiShelfStateChange	.1.3.6.1.4.1.529.0.25	✓	-	-	-	🟢	🔧	🔧	Customer	multiShelfStateChange
multicastHeartBeatMonitor	.1.3.6.1.4.1.529.0.19	✓	-	-	-	🟢	🔧	🔧	Customer	multicastHeartBeatMonitor
portAcrPiding	.1.3.6.1.4.1.529.0.10	✓	-	-	-	🟢	🔧	🔧	Customer	portAcrPiding
portCarrier	.1.3.6.1.4.1.529.0.8	✓	-	-	-	🟢	🔧	🔧	Customer	portCarrier
portCollectDigits	.1.3.6.1.4.1.529.0.5	✓	-	-	-	🟢	🔧	🔧	Customer	portCollectDigits
portConnected	.1.3.6.1.4.1.529.0.7	✓	-	-	-	🟢	🔧	🔧	Customer	portConnected
portDteNotReady	.1.3.6.1.4.1.529.0.11	✓	-	-	-	🟢	🔧	🔧	Customer	portDteNotReady
portDualDelay	.1.3.6.1.4.1.529.0.1	✓	-	-	-	🟢	🔧	🔧	Customer	portDualDelay
portHaveSerial	.1.3.6.1.4.1.529.0.3	✓	-	-	-	🟢	🔧	🔧	Customer	portHaveSerial
portInactive	.1.3.6.1.4.1.529.0.0	✓	-	-	-	🟢	🔧	🔧	Customer	portInactive
portLoopback	.1.3.6.1.4.1.529.0.9	✓	-	-	-	🟢	🔧	🔧	Customer	portLoopback
portRinging	.1.3.6.1.4.1.529.0.4	✓	-	-	-	🟢	🔧	🔧	Customer	portRinging
portUseExceeded	.1.3.6.1.4.1.529.0.13	✓	-	-	-	🟢	🔧	🔧	Customer	portUseExceeded
portWaitSerial	.1.3.6.1.4.1.529.0.2	✓	-	-	-	🟢	🔧	🔧	Customer	portWaitSerial
portWaiting	.1.3.6.1.4.1.529.0.6	✓	-	-	-	🟢	🔧	🔧	Customer	portWaiting
powerSupplyOperationalStateChange	.1.3.6.1.4.1.529.0.24	✓	-	-	-	🟢	🔧	🔧	Customer	powerSupplyOperationalStateChange
powerSupplyStateChange	.1.3.6.1.4.1.529.0.23	✓	-	-	-	🟢	🔧	🔧	Customer	powerSupplyStateChange
radiusServerChange	.1.3.6.1.4.1.529.0.18	✓	-	-	-	🟢	🔧	🔧	Customer	radiusServerChange

Updated: 8/13/14 08:10:51 AM      Total: 45      Selected: 0      Filter: ON      Auto refresh: OFF

1. Edit the trap configuration to include the source node and the port in the format for easier reading. Do this for all 3 trap mentioned above – an example snapshot is shown in following figure.

### SNMP Trap Configuration: Message Format

The screenshot shows the 'SNMP Trap Configuration' window. The 'Message Format' field is highlighted with a red circle and contains the text: `portConnected SourceNode=$snr Port=$1.3.6.1.2.1.2.2.1.1`. The 'Name' field is 'portConnected' and the 'SNMP Object ID' is '.1.3.6.1.4.1.529.0.7'. The 'Enabled' checkbox is checked. The 'Category' is 'Status', 'Family' is 'Node', and 'Severity' is 'Normal'. The 'Description' field contains a detailed explanation of the trap and its variables.

Here is what the three trap look like when they are sent:

```
# nnmsnmnotify.ovpl -v 1 -a ciscope6524.ind.hp.com localhost
.1.3.6.1.4.1.529.0.6 .1.3.6.1.2.1.2.2.1.1 integer 1

# nnmsnmnotify.ovpl -v 1 -a ciscope6524.ind.hp.com localhost
.1.3.6.1.4.1.529.0.7 .1.3.6.1.2.1.2.2.1.1 integer 1

# nnmsnmnotify.ovpl -v 1 -a ciscope6524.ind.hp.com localhost
.1.3.6.1.4.1.529.0.8 .1.3.6.1.2.1.2.2.1.1 integer 1
```

### SNMP Traps

Sever	Lifec	Last Occurrence	Time	Source Node	Source Object	Cate	Famil	Corr	Message	Notes
✓		8/13/14 8:20:41 AM		ciscope6524	none				portCarrier SourceNode=ciscope6524 Port=1	
✓		8/13/14 8:20:36 AM		ciscope6524	none				portConnected SourceNode=ciscope6524 Port=1	
✓		8/13/14 8:20:31 AM		ciscope6524	none				portWaiting SourceNode=ciscope6524 Port=1	

2. Build two pairwise correlations.
  - a. Select the Second Incident Configuration as portConnected and the First Incident Configuration as portCarrier. You do not need to define any payload filters but you must define matching criteria. Remember that there is a built-in criterion for Source Node. In addition, you must match against the port as defined by the varbind `.1.3.6.1.2.1.2.2.1.1`. You could also use `$1` but it is usually better to use explicit OIDs rather than position numbers. Note that the duration has been set to zero. This means that NNMi searches back through the entire incident database to find the match, but after it finds a match, it stops correlating. This results in a single pairing. This may not be desired but is shown as an example here.



## Pairwise Configuration: ascendPortCarrierConnectedPair

Pairwise Configurations Pairwise Configuration \*

Save and Close Delete Pairwise Configuration

**Basics**

Name: ascendPortCarrierConnectedPair

Enabled: ☒

First Incident Configuration: portConnected

Second Incident Configuration: portCarrier

Description:

Author: Customer

Duration: 0.00 Seconds

Delete When Canceled: ☐

**Matching Criteria**

Specify the Matching Criteria only if you want to use Matching Criteria in addition to the following criteria that NMI uses automatically:

SNMP Trap Incidents: c/a address of the source address for the trap  
 Management Events Incidents: Name of the Source Object and Source Node  
 Remote NIM 6.x/7.x Event Incidents: c/a remotemgr (IP Address or Hostname) of the NMI management station sending the incident  
 and c/a address of the source address for the trap  
 Syslog Message Incidents: none

See the "Matching Criteria Configuration Form (Identify Incident Pairs)" help topic for more information.

First Incident Criteria: 1.3.6.1.2.2.1.1

Second Incident Criteria: 1.3.6.1.2.2.1.1

## Pairwise Configuration: ascendPortCarrierWaitingPair

Pairwise Configurations Pairwise Configuration \*

Save and Close Delete Pairwise Configuration

**Basics**

Name: ascendPortCarrierWaitingPair

Enabled: ☒

First Incident Configuration: portWaiting

Second Incident Configuration: portCarrier

Description:

Author: Customer

Duration: 0.00 Seconds

Delete When Canceled: ☐

**Matching Criteria**

Specify the Matching Criteria only if you want to use Matching Criteria in addition to the following criteria that NMI uses automatically:

SNMP Trap Incidents: c/a address of the source address for the trap  
 Management Events Incidents: Name of the Source Object and Source Node  
 Remote NIM 6.x/7.x Event Incidents: c/a remotemgr (IP Address or Hostname) of the NMI management station sending the incident  
 and c/a address of the source address for the trap  
 Syslog Message Incidents: none

See the "Matching Criteria Configuration Form (Identify Incident Pairs)" help topic for more information.

First Incident Criteria: 1.3.6.1.2.2.1.1

Second Incident Criteria: 1.3.6.1.2.2.1.1

- b. Receive the first two traps. Nothing is correlated.

## SNMP Traps: Traps Received

Sever	Life	Last Occurrence	Source Node	Source Object	Categ	Famil	Corr	Message	Notes
✓	8/13/14 8:41:44 AM	ciscope6524	none					portConnected SourceNode=ciscope6524 Port=1	
✓	8/13/14 8:41:39 AM	ciscope6524	none					portWaiting SourceNode=ciscope6524 Port=1	





















- c. When you receive the portCarrier trap, it closes the other two traps.

## SNMP Traps: portCarrierTrap Received

Sever	Life	Last Occurrence	Source Node	Source Object	Categ	Famil	Corr	Message	Notes
✓	8/13/14 8:43:38 AM	ciscope6524	none					portCarrier SourceNode=ciscope6524 Port=1	
✓	8/13/14 8:41:44 AM	ciscope6524	none					portConnected SourceNode=ciscope6524 Port=1	
✓	8/13/14 8:41:39 AM	ciscope6524	none					portWaiting SourceNode=ciscope6524 Port=1	

- d. Because the duration is set to zero, only the first pairings are closed. If there were other traps in the database that had potential matches, as shown below, those traps are not closed

## SNMP Traps: Other Traps with Potential Matches

SNMP Traps										
Severity	Life Cycle	Last Occurrence	Time	Source Node	Source Object	Category	Family	Correct	Message	Notes
✓		8/13/14 8:43:38 AM		ciscope6524	none				portCarrier SourceNode=ciscope6524 Port=1	
✓		8/13/14 8:41:44 AM		ciscope6524	none				portConnected SourceNode=ciscope6524 Port=1	
✓		8/13/14 8:41:39 AM		ciscope6524	none				portWaiting SourceNode=ciscope6524 Port=1	
✓		8/13/14 8:20:36 AM		ciscope6524	none				portConnected SourceNode=ciscope6524 Port=1	
✓		8/13/14 8:20:31 AM		ciscope6524	none				portWaiting SourceNode=ciscope6524 Port=1	

## Batch Incident Configuration

NNMi 10.00 allows you to make modifications to collections of incident configurations using a file. This facilitates the process of making many similar changes without having to use the NNMi console for each incident configuration. For example, suppose you want to add the same action to a set of incidents. With batch incident configuration, you can export the current configuration, modify the file, and then import the modified configuration in just a few simple steps.

There are two new tools introduced in NNMi 9.20 which are available in NNMi 10.00 as well: `nnmincidentcfgdump.ovpl` and `nnmincidentcfgload.ovpl`. You use `nnmincidentcfgdump.ovpl` to export the current configuration and you use `nnmincidentcfgload.ovpl` to import the new configuration. These tools use a tag formatted file (not XML or free-form text).

This document does not give the complete syntax for these tools. There are a number of very good examples in the following directory:

- *UNIX:* `/opt/OV/examples/nnm/incidentcfg`
- *Windows:* `<drive>\Program Files (x86)\HP\HP BTO Software\examples\nnm\incidentcfg`

One of the easiest ways to use the `nnmincidentcfgload.ovpl` tool is to generate an example and compare the differences. Then you can carry these differences over to other incidents. Consider the following example.

Suppose you want to load the F5-BIGIP-COMMON-MIB file to get some new traps defined in NNMi for F5 BIG-IP.

1. Load the MIB with the `nnmloadmib.ovpl` command:
 

```
# nnmloadmib.ovpl -load F5-BIGIP-COMMON-MIB.mib
Successfully completed operation LoadMib.
16 MIB Variables were loaded.
134 Traps were loaded.
```
2. Load the traps into the trap configuration using the `nnmincidentcfg.ovpl` command.
 

```
# nnmincidentcfg.ovpl -loadTraps F5-BIGIP-COMMON-MIB
SNMP trap(s) from mib module loaded: F5-BIGIP-COMMON-MIB. Number of traps:
134.
The following traps were added to incident configuration: bigipNodeUp -
.1.3.6.1.4.1.3375.2.4.0.13
bigipAgentStart - .1.3.6.1.4.1.3375.2.4.0.1
bigipStandby - .1.3.6.1.4.1.3375.2.4.0.14
... (many traps not shown here)
bigipDiskPartitionGrowth - .1.3.6.1.4.1.3375.2.4.0.26
bigipAsmRequestBlocked - .1.3.6.1.4.1.3375.2.4.0.38
bigipGtmAppAvail - .1.3.6.1.4.1.3375.2.4.0.71
```

Suppose you want to apply an action to all of these traps. Because there are 134 traps, it would take much effort to open each trap configuration individually and add the action. So you will take advantage of the batch configuration.

3. Before going any further, it is a good idea to export the incident configuration before making changes. This allows you to revert to this same place using the `nnmconfigimport.ovpl` command, if necessary. (Alternatively, you could perform a backup of NNMi.)

```
# nnmconfigexport.ovpl -c incident -f /var/tmp Successfully exported
/var/tmp/incident.xml.
```

4. Dump one trap to provide an example. Use the trap `bigipAgentStart` and specify the dump command to only dump this trap configuration based on the OID `.1.3.6.1.4.1.3375.2.4.0.1`.

```
# nnmincidentcfgdump.ovpl -dump bigipAgentStart_before.tag -oid
.1.3.6.1.4.1.3375.2.4.0.1
Starting a user transaction with a timeout of: 3,600 seconds.
```

**Here is the file it created:**

```
# cat bigipAgentStart_before.tag
*ConfigurationType=SnmpTrapConfig
```

```

*Name bigipAgentStart
*Oid .1.3.6.1.4.1.3375.2.4.0.1

-Author
    -Key com.customer.author
-Category
    -Key com.hp.nms.incident.category.Status
-Enable true

-ActionConfiguration
    -Enable false
-DampenConfiguration
    -Enable false
    -HourInterval 0
    -MinuteInterval 0
    -SecondInterval 0

-DedupConfiguration
    -ComparisonCriteria NAME
    -DedupCount 2
    -Enable false
    -HourInterval 0
    -MinuteInterval 0
    -SecondInterval 0
-Description An indication that the agent has started running.
-Family
    -Key com.hp.nms.incident.family.Node

-GeoCentralForwardConfiguration
    -Enable false

-MessageFormat bigipAgentStart
-Severity NORMAL
-EnrichConfiguration
    -Enable false
-SuppressConfiguration
    -Enable false

-RateConfiguration
    -ComparisonCriteria NAME
    -Enable false
    -HourInterval 0
    -MinuteInterval 0
    -RateCount 0
    -SecondInterval 0
-UserRootCause false

```

5. Go into the NNMi console and add the action to this trap configuration. Save the trap configuration after making the change.

## SNMP Trap Configuration: Actions

The screenshot shows the 'SNMP Trap Configuration' window. The 'Name' field is 'bigipAgentStart'. The 'SNMP Object ID' is '.1.3.6.1.4.1.3375.2.4.0.1'. The 'Enabled' checkbox is checked. The 'Category' is 'Status', 'Family' is 'Node', and 'Severity' is 'Normal'. The 'Message Format' is 'bigipAgentStart'. The 'Description' is 'An indication that the agent has started running.' The 'Author' is 'Customer'. The 'Actions' tab is selected, showing a table with one row: 'ScriptOrExecutable' with command '/var/opt/OV/shared/nnm/actions/bigIPAction.ksh \$name \$snn'.

6. Dump the configuration again with the action added.

```
# nmmincidentcfgdump.ovpl -dump bigipAgentStart_after.tag -oid
.1.3.6.1.4.1.3375.2.4.0.1
Starting a user transaction with a timeout of: 3,600 seconds.
```

Here is the file it created with the newly added lines highlighted. Notice the relatively simple format that was added.

```
# cat bigipAgentStart_after.tag

*ConfigurationType=SnmpTrapConfig
  *Name bigipAgentStart
  *Oid .1.3.6.1.4.1.3375.2.4.0.1
  -Author
    -Key com.customer.author
  -Category
    -Key com.hp.nms.incident.category.Status
  -Enable true
  -ActionConfiguration
    -Enable true
    -Actions
      -Action
        -Command
          /var/opt/OV/shared/nnm/actions/bigIPAction.ksh
          $name $snn
        -CommandType SCRIPT_OR_EXECUTABLE
        -LifecycleState Registered
  -DampenConfiguration
    -Enable false
    -HourInterval 0
    -MinuteInterval 0
    -SecondInterval 0
  -DedupConfiguration
    -ComparisonCriteria NAME
    -DedupCount 2
    -Enable false
    -HourInterval 0
    -MinuteInterval 0
    -SecondInterval 0
  -Description An indication that the agent has started running.
  -Family
    -Key com.hp.nms.incident.family.Node
```

```

-GeoCentralForwardConfiguration
  -Enable false
-MessageFormat bigipAgentStart
-Severity NORMAL
-EnrichConfiguration
  -Enable false
-SuppressConfiguration
  -Enable false
-RateConfiguration
  -ComparisonCriteria NAME
  -Enable false
  -HourInterval 0
  -MinuteInterval 0
  -RateCount 0
  -SecondInterval 0
-UserRootCause false

```

- Now that you have learned the format of the lines you must add, dump all 134 traps. You can dump the whole family of traps using an OID wildcard .1.3.6.1.4.1.3375.\*.

```
# nnmincidentcfgdump.ovpl -dump bigip_before.tag -oid .1.3.6.1.4.1.3375.*
Starting a user transaction with a timeout of: 3,600 seconds.
```

- Edit this file and add the highlighted lines to all the traps. Save the file as bigip\_after.tag, and then load this file into NNMi using nnmincidentcfgload.ovpl as shown.

```
# nnmincidentcfgload.ovpl -load bigip_after.tag Translated: 134
configurations, now attempting import.
```

Starting a user transaction with a timeout of: 3,600 seconds.

- Go to any of the BIG-IP traps in the NNMi console and notice that the action has been added.

### SNMP Trap Configuration: Action Added

SNMP Trap Configurations | SNMP Trap Configuration

Save and Close | Delete SNMP Trap Configuration

**Basics**

For information about troubleshooting incidents, click [here](#).

Name: bigipBladeOffline

The SNMP Object ID (OID) attribute accepts one wildcard character (\*) that must appear at the end of the OID specified. NNMi permits wildcards only in OIDs beginning with .1.3.6.1.4 (private MIBs). Click [here](#) for more information.

SNMP Object ID: .1.3.6.1.4.1.3375.2.4.0.90

Enabled: ☒

Root Cause: ☐

Category: Status

Family: Node

Severity: Normal

Specify how the incident message appears in the incident view. To include incident information in the message use \$(variable\_name). Select these variables from a set of valid parameters or Custom incident attributes. For more information, click [here](#).

Message Format: bigipBladeOffline

Description: A blade has failed - offline.  
Variables:  
1: bigipNotifyObjMsg  
Syntax: TEXTUAL\_CONVENTION  
Description: The additional information about the related notification.

Author: Customer

**Actions**

You configure actions to automatically run at any point in the Incident lifecycle. For example, when an Incident is generated (Registered), you can automatically open a trouble ticket, send email, or page your network operator. NNMi supports running a Jython file, executable, or script.

Note: Your configured actions are disabled until you click Enabled and Save this form.

Enabled: ☒

**Lifecycle Transition Actions**

*	LifeCycle	Command Type	Command
	ScriptOrExecutable		/var/opt/OV/shared/nnm/actions/bigipAction.ksh \$name \$ann

You have now added an action to all 134 traps at one time.

If you wanted to revert back to the earlier configuration, there are two ways you could do this. First, you could import the configuration snapshot that you took earlier as shown here:

```
# nnmconfigimport.ovpl -f incident.xml We have sorted the list like:
incident, Successfully imported incident.xml.
```



Or you could load the configuration of the earlier file, which represents the traps before you made your modifications:

```
# nnmincidentcfgload.ovpl -load bigip_before.tag Translated: 134
configurations, now attempting import.
Starting a user transaction with a timeout of: 3,600 seconds.
```

## Legal Notices

### Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

### Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

### Copyright Notices

© Copyright 2009–2014 Hewlett-Packard Development Company, L.P.

### Trademark Notices

Adobe® is a trademark of Adobe Systems Incorporated.

HP-UX Release 10.20 and later and HP-UX Release 11.00 and later (in both 32 and 64-bit configurations) on all HP 9000 computers are Open Group UNIX 95 branded products.

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

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

### Oracle Technology — Notice of Restricted Rights

Programs delivered subject to the DOD FAR Supplement are 'commercial computer software' and use, duplication, and disclosure of the programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, programs delivered subject to the Federal Acquisition Regulations are 'restricted computer software' and use, duplication, and disclosure of the programs, including documentation, shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software-Restricted Rights (June 1987). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

For the full Oracle license text, see the license-agreements directory on the NNMi product DVD.

### Acknowledgements

This product includes software developed by the Apache Software Foundation.

(<http://www.apache.org>)

This product includes software developed by the Indiana University Extreme! Lab.

(<http://www.extreme.indiana.edu>)

### Support

Visit the HP Software Support web site at:

**[www.hp.com/go/hpsoftwaresupport](http://www.hp.com/go/hpsoftwaresupport)**

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract.

To register for an HP Passport ID, go to:

**<http://h20229.www2.hp.com/passport-registration.html>**

To find more information about access levels, go to:

[http://h20230.www2.hp.com/new\\_access\\_levels.jsp](http://h20230.www2.hp.com/new_access_levels.jsp)

**Sign up for updates**

<http://h20230.www2.hp.com/selfsolve/manuals>

---

© Copyright 2009 – 2014 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.



# We appreciate your feedback!

If you have comments about this document, you can [contact the documentation team by email](#). If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line: Feedback on White Paper (Network Node Manager i Software 10.00)  
Just add your feedback to the email and click send. If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to [docfeedback@hp.com](mailto:docfeedback@hp.com).