HP Network Node Manager i Software

Step- by-Step Guide to Incident Management

Release 9.23

This whitepaper describes the NNMi event pipeline and Incident configuration. It includes the following Incident configuration options:

- Deduplication
- Rate Correlation
- Incident Suppression
- Enrichment
- Actions

It also includes information about Dampening incidents and explains how to narrow incident customization based on Node Group and Interface Group membership.
This whitepaper uses the following terms:

Trap - an asynchronous notification from an SNMP agent on a managed node that is sent to the NNMi management server.

NNMi management event - an incident that is generated by NNMi usually as a result of a status poll. An example is the Node Down incident.

Also see the "Step-by-Step Guide to Managing SNMP Traps in NNMi".

Note: Your NNMi console appearance might vary from some of the figures in this document.

CONTENTS

Setting Up Your SNMP Trap

In this example scenario, a network device sends the same example SNMP trap to mean various things. The difference between the traps is the varbinds. This is a common practice for some devices and applications.

Note: The examples presented are based on traps, but the same principles apply to management events such as Node Down.

The trap and its varbinds are defined below:

OID .1.3.6.1.4.1.33333.0.1
Varbind 1: .1.3.6.1.4.1.33333.1.1.1 (Integer)
Varbind 2: .1.3.6.1.4.1.33333.1.2.1 (Octet String)

The following table describes the Varbind 1 and Varbind 2 values:

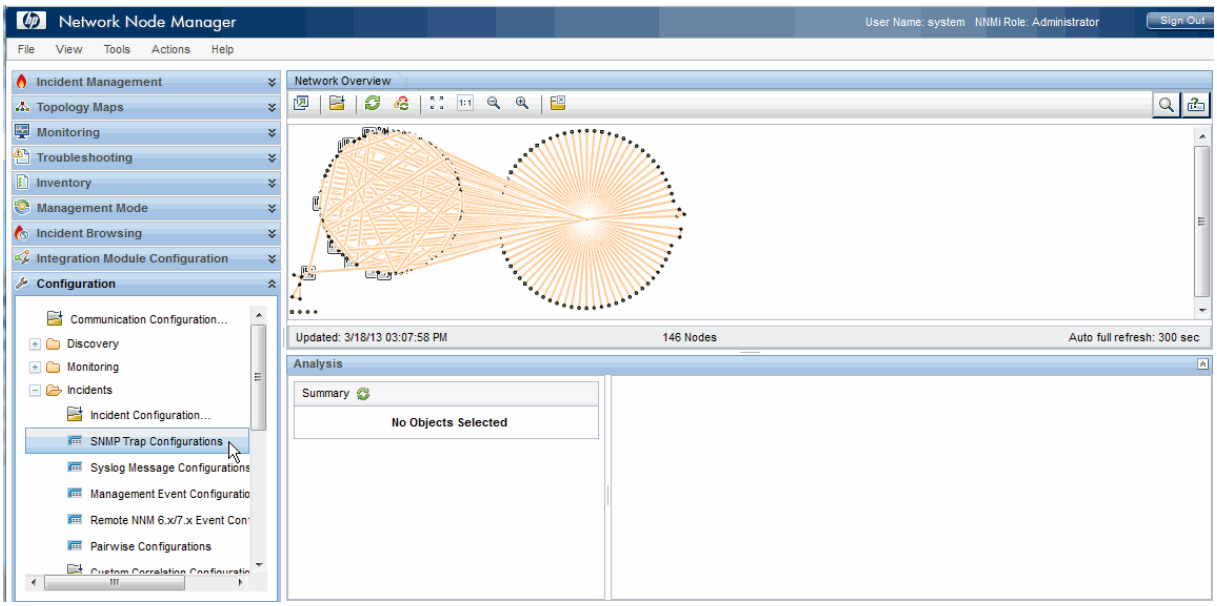
Varbind1 Type: Integer	**Varbind1 Type Description**: Status
Varbind1 Value	**Description**
1	Normal Status
2	Warning Status
3	Critical Status
Varbind2 Type: String	**Varbind1 Type Description**: Module with problem
Varbind2 Value	**Description**
CPU	CPU is the source of the problem
Temperature	Temperature is the source of the problem

Because this trap does not exist, no MIB defines the trap. Therefore, this example begins by creating the trap definition. However, normally, to begin, load the trap definition using the following command:

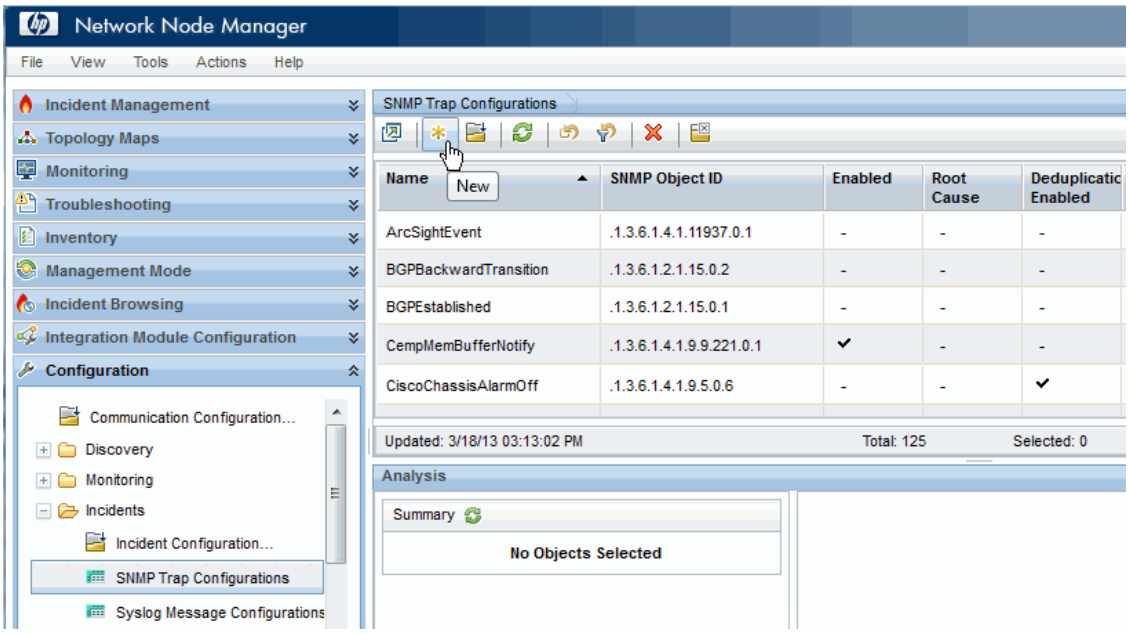
```
nnmincidentcfg.ovpl –loadTraps <mib_file>
```

To create the trap definition:

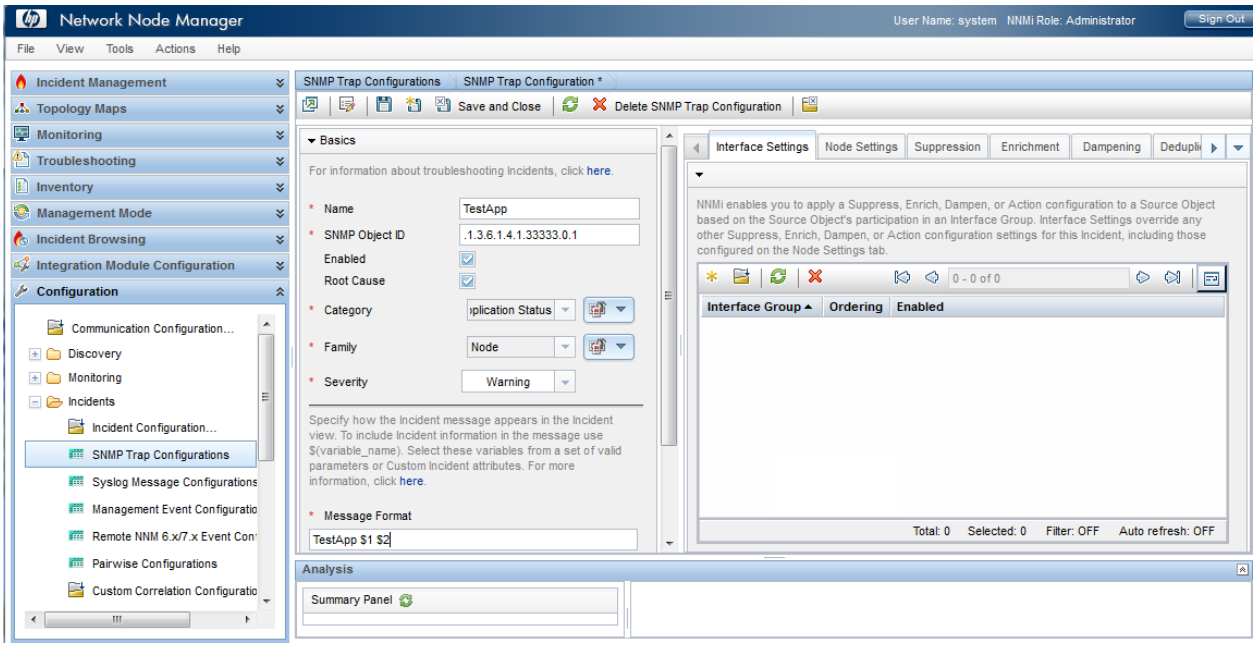
1. Navigate to the **Configuration** workspace, expand **Incidents**, and click **SNMP Trap Configurations**.



2. Navigate to the **SNMP Traps** tab and select the * icon.



3. In the **Name** attribute, enter **TestApp**.
4. In the **SNMP Object ID** attribute, enter **.1.3.6.1.4.1.33333.0.1**
5. Click **Enabled**.
6. Click **Root Cause** so that these traps will display in the Key Incidents view.
7. In the **Category** attribute, select **Application Status**.
8. In the **Family** attribute, select **Node**.
9. In the **Severity** attribute, select **Warning**.
10. In the **Message Format** attribute, enter "TestApp $1 $2".
11. In the **Author** attribute, select Customer.
12. Click **Save and Close** to save the changes.



13. Next, use the `nnmsnmpnotify.ovpl` command to send the example traps:

```
# nnmsnmpnotify.ovpl -a 15.2.127.135 localhost .1.3.6.1.4.1.33333.0.1 .1.3.6.1.4.1.33333.1.1.1
integer 2  .1.3.6.1.4.1.33333.1.2.1 OCTETSTRING CPU
# nnmsnmpnotify.ovpl -a 15.2.121.254 localhost .1.3.6.1.4.1.33333.0.1 .1.3.6.1.4.1.33333.1.1.1
integer 1  .1.3.6.1.4.1.33333.1.2.1 OCTETSTRING Temperature
```

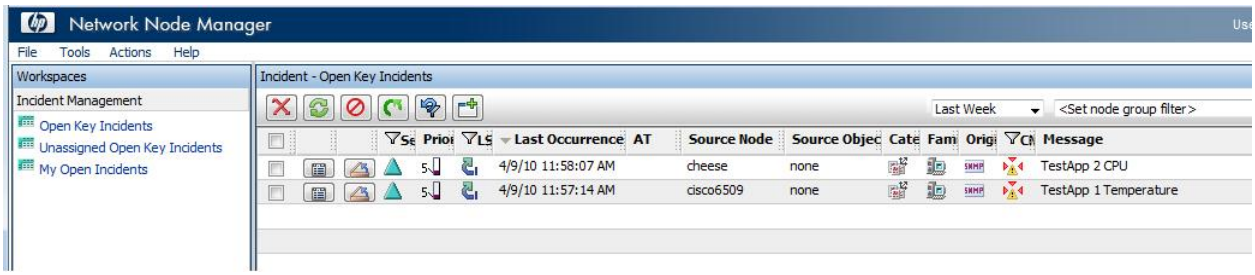
Note the following:

- This command must be run from the NNMi server.
- Each `nnmsnmpnotify.ovpl` command is a single line.

To confirm that NNMi has received the traps:

1. Navigate to the **Incident Browsing** workspace.
2. Click **Open Key Incidents** to confirm that the traps have been received.

Note: Use the pull-down menu to change the time period if necessary.

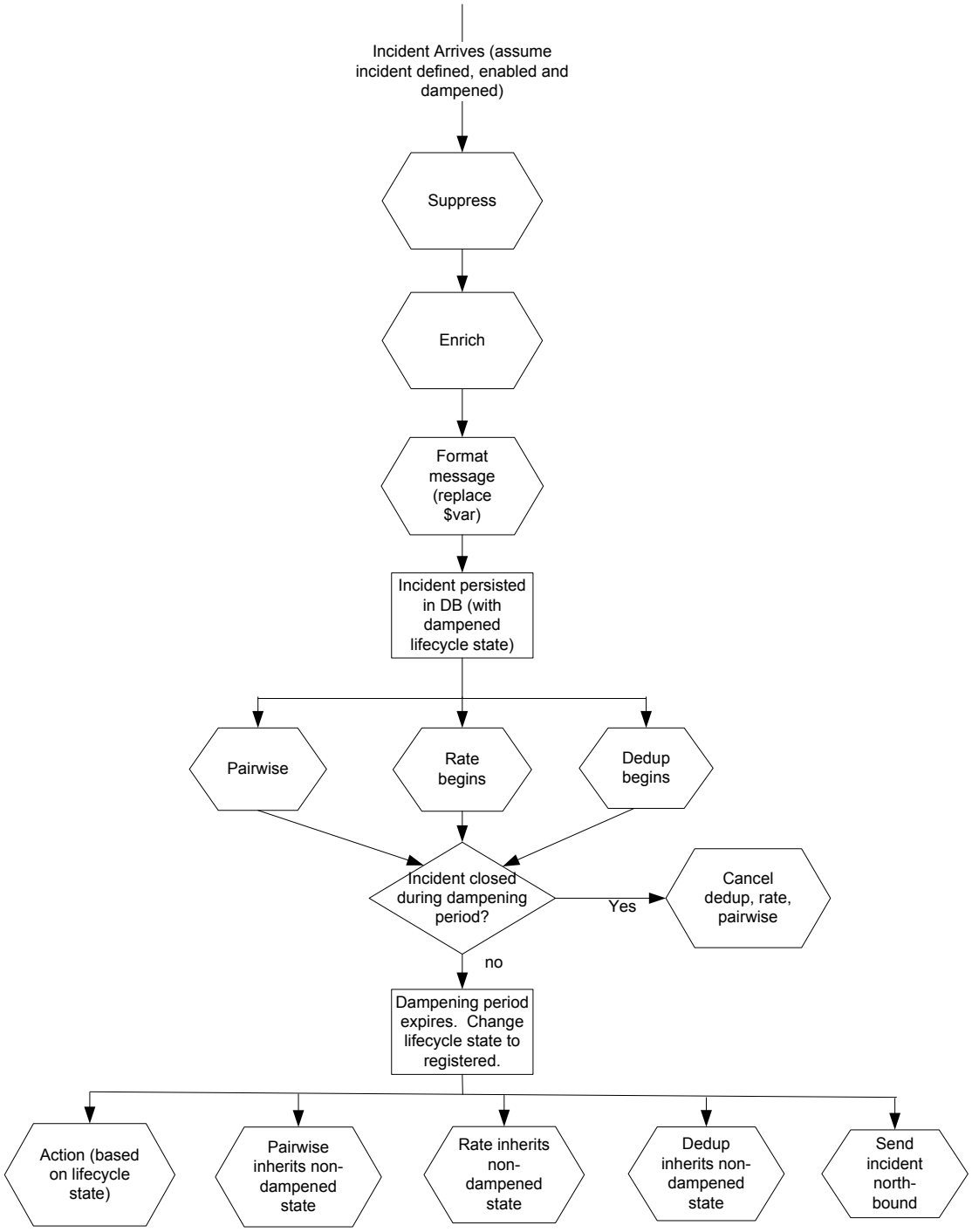


Now you are ready to begin working with these traps.

Dampening and the Incident Pipeline

NNMi provides the Incident Dampening feature to enable you to ignore network "noise" in which interfaces and nodes are down for short periods of time. With Incident Dampening, NNMi behaves as if the short outages never occurred. To identify these incidents, NNMi uses the **Dampened** Lifecycle State. The **Dampened** Lifecycle State precedes the **Registered** Lifecycle State.

The following flowchart provides a summary of where Dampening fits into the NNMi Incident Pipeline.



As shown in the preceding flowchart, when an incident arrives, NNMi checks whether the incident can be suppressed and immediately discarded. If an incident is not suppressed, NNMi determines whether an Enrichment Configuration is enabled for the incident. (Enrichment Configuration is used to customize a subset of incident configuration attributes, such as Message Format or Priority.)
Next, NNMi replaces any parameter strings (for example $sourceNodeName) specified in the Message Format.

If Dampening is enabled for the incident, NNMi sets the Lifecycle State to Dampened. If Dampening is not enabled for the incident, NNMi sets the Lifecycle State to Registered.
After the Lifecycle State is set, Rate and Deduplication correlations, as well as Pairwise matching takes place.

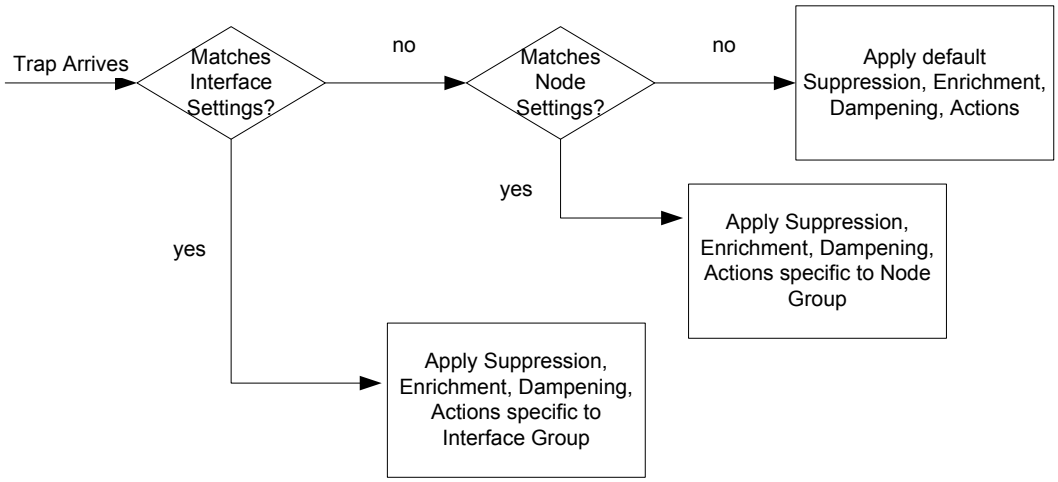
If NNMi cancels the incident and sets the Lifecycle State to Closed during this Dampening period, NNMi discards the incident and discontinues any Rate or Deduplication correlation and Pairwise matching. If the Dampening period expires, NNMi sets the Lifecycle State to Registered and continues any Rate or Deduplication correlation and Pairwise matching.

Customizing Incident Configurations Using Interface or Node Groups

Incidents can be customized based on Interface Groups or Node Groups. This feature is reflected in the incident configuration form. When a trap arrives into the Incident Pipeline (after it has cleared any filtering), NNMi compares the SNMP trap to the Interface Settings to see if the source of the trap is a member of this Interface Settings group. If NNMi finds a match on the source interface (source object), NNMi applies the Suppression, Enrichment, Dampening, and Actions specific to that Interface Group.

Note: If one of these tabs is disabled (for example, Enrichment), NNMi does not use the default Enrichment.

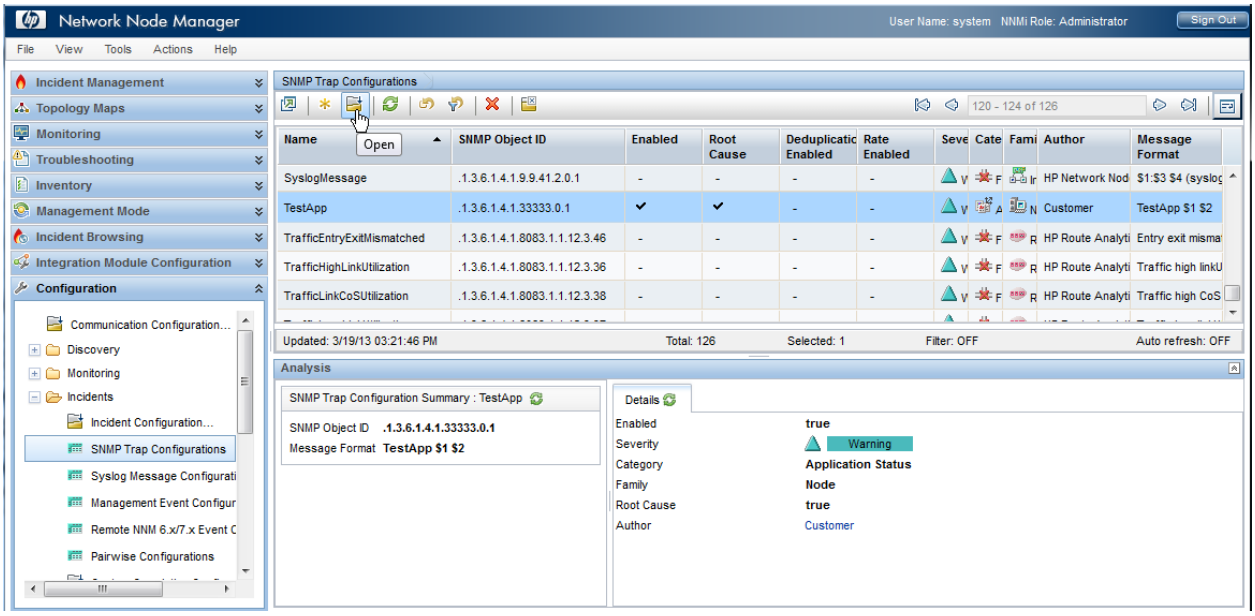
If the source interface does not match any Interface Settings, NNMi compares the source node to the Node Settings group. If NNMi finds a match on the source node, NNMi applies the Suppression, Enrichment, Dampening, and Actions specific to that Node Group.
If the source node does not match any Node Settings, then the default Suppression, Enrichment, Dampening and Actions are applied.



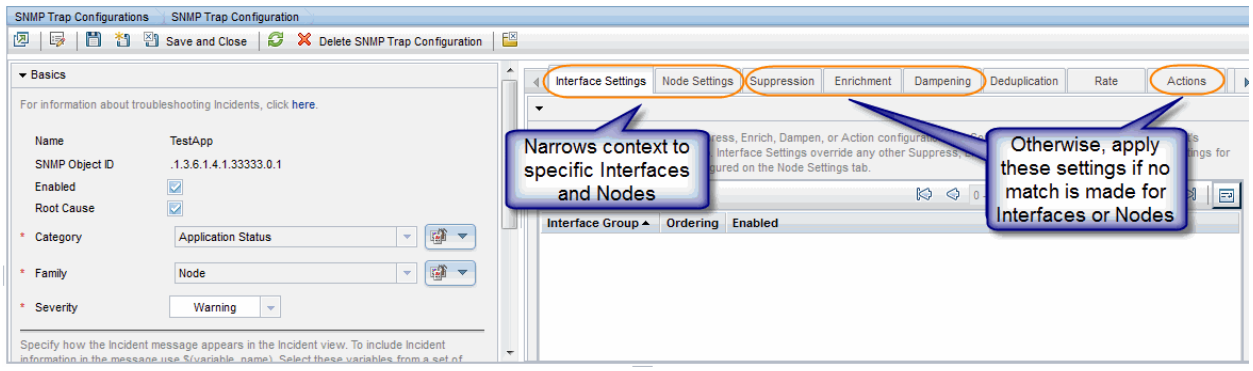
NNMi applies Deduplication and Rate Correlation independent of Interface Settings and Node Settings.

To view the Incident Configuration options for Interface and Node Settings:

1. Navigate to the **Configuration** Workspace.
2. Open **Incidents** and select **SNMP Trap Configurations**.
3. Locate and select the TestApp trap, and then click the Open button.



The graphic below indicates how the various tabs apply to this concept.

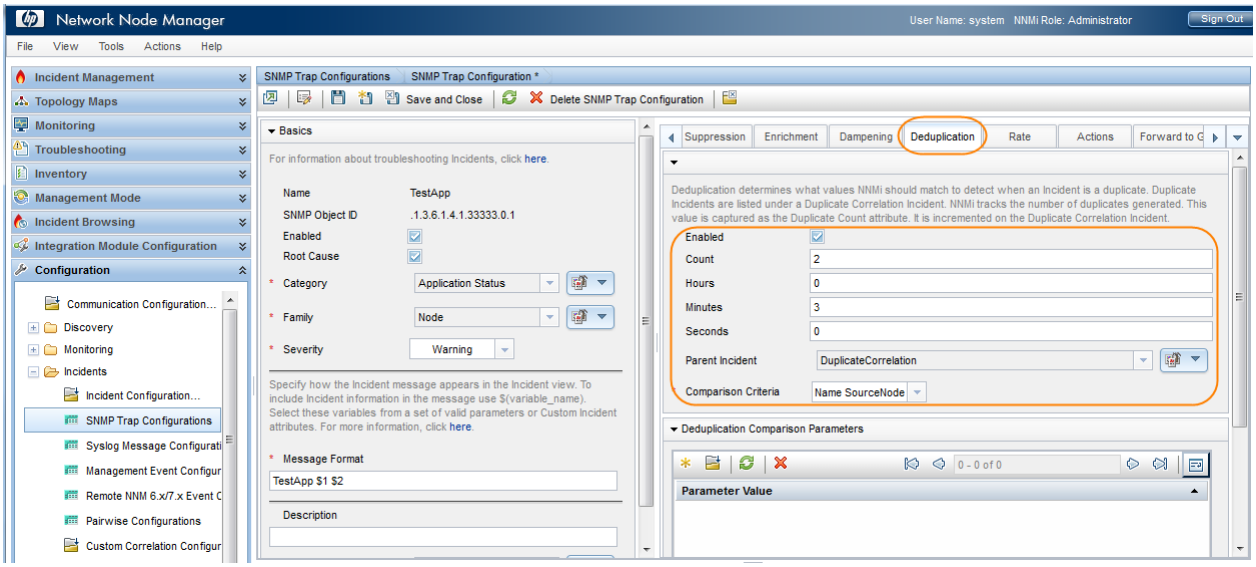


Deduplication

NNMi's Deduplication feature enables you to correlate duplicate incidents under a new incident. It also deletes duplicate incidents once a specified number of duplicate incidents are generated.
This example configures Deduplication for the TestApp SNMP trap configuration.

1. Navigate to the **Configuration** Workspace.
2. Open **Incidents** and select **SNMP Trap Configurations**.
3. Locate and select the TestApp trap, and then click the Open button.
4. Click to check **Enabled**.

5. Navigate to the **Deduplication** tab.
6. In the **Count** attribute, enter the number of TestApp traps that NNMi should retain in the database for a particular Deduplication time period. The maximum number is 10. For this example, enter **2**.
7. In the **Hours**, **Minutes**, and **Seconds** attributes, specify the time that must elapse before a new duplicate incident is generated for this incident configuration. For this example, in the Minutes attribute, enter **3**.
8. Next, in the **Parent Incident** drop-down list, select the incident that you want NNMi to generate to indicate that Deduplication has occurred. For this example, select **DuplicateCorrelation**.
9. Last, specify the **Comparison Criteria**. The Comparison Criteria specifies what attributes NNMi should use to decide what constitutes a duplicate. This example uses Name and SourceNode. This means that when two TestApp SNMP traps arrive, NNMi considers them to be duplicate if the SNMP traps have the same name (TestApp) and the same SourceNode (came from the same device in the network).

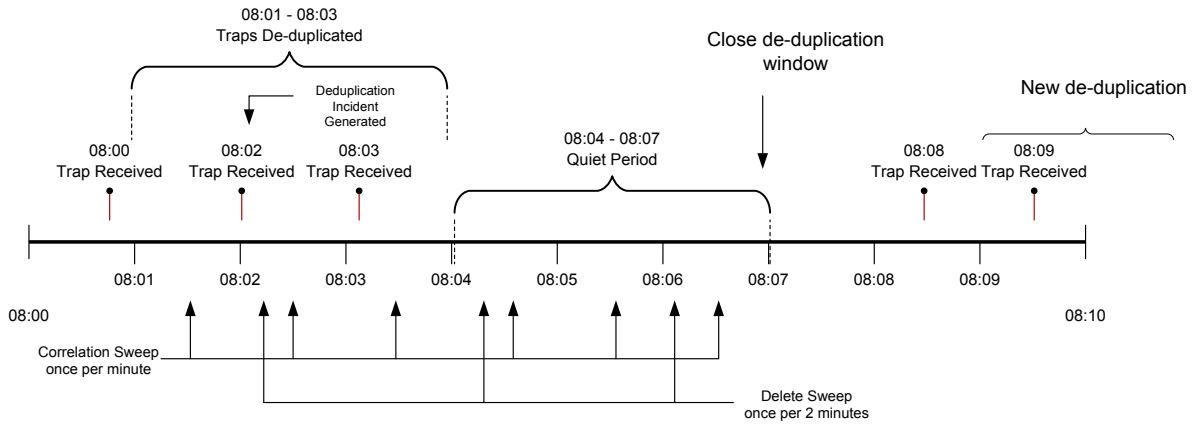


The following diagram depicts the following scenario:

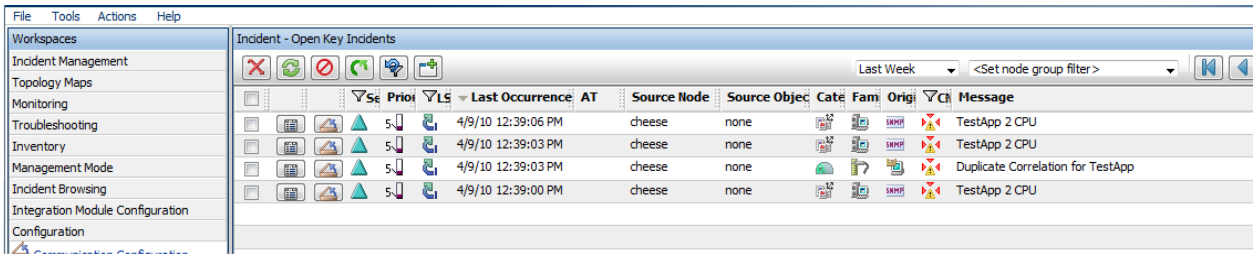
The first TestApp SNMP trap arrives at 8:00. Another trap with the same name and source node arrives at 8:02. NNMi generates a new DuplicateCorrelation incident. At 8:03 another trap arrives. In addition, every minute NNMi sweeps the incidents to determine whether to correlate duplicate incidents. At approximately 8:02:30, NNMi correlates the first two SNMP traps under the DuplicationCorrelation incident and marks them as Correlated Children. At 8:03:30, NNMi correlates the third SNMP trap as a child to the DuplicationCorrelation incident. At approximately 8:04:15, NNMi checks whether more than two TestApp SNMP traps are correlated under a single DuplicationCorrelation. NNMi deletes one of the TestApp SNMP traps because the total number is three.

Note: Although NNMi deletes the third SNMP trap from the NNMi database, the total count of **3** is retained in the DuplicateCorrelation incident as the Duplicate Count.

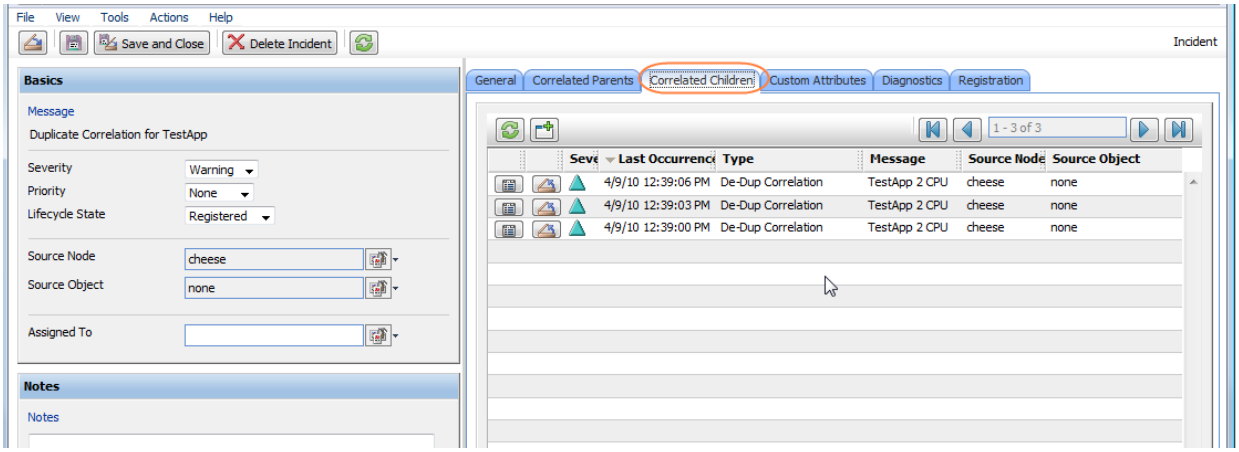
After SNMP traps stop arriving for three minutes (the time window for this Deduplication), NNMi closes the Deduplication time window. At 8:08 a new TestApp trap arrives. At 8:09 another TestApp SNMP trap arrives from the same node. This begins the cycle again. NNMi generates a new DeduplicationCorrelation incident and continues to evaluate each incident as previously described.



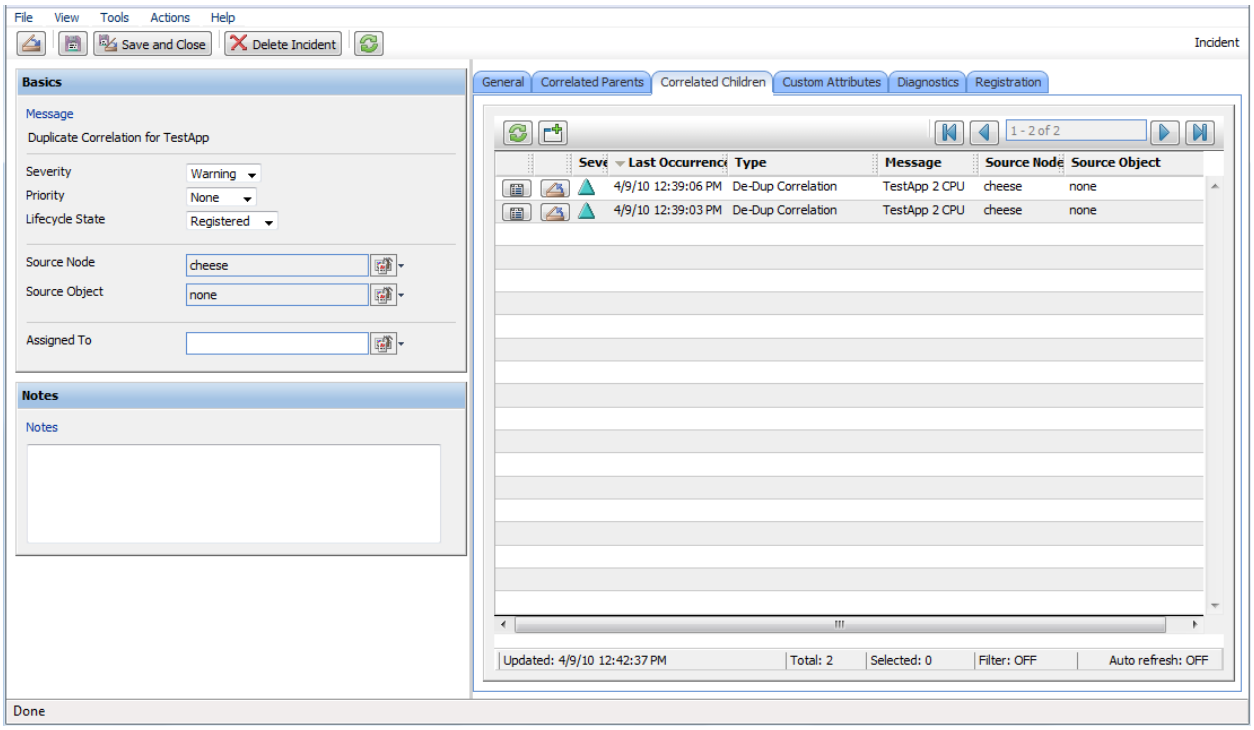
The following image depicts how these incidents might appear in the Open Key Incidents view. Note that the times in the example below do not correspond exactly with the timeline above, but the order is similar. As shown in the example, three traps arrived and a new DuplicateCorrelation incident is generated.



When NNMi sweeps the incidents to determine whether to correlate duplicate incidents, it correlates the three traps under the DuplicateCorrelation incident as shown in the following example:



Next, NNMi checks whether more than two TestApp SNMP traps are correlated under a single DuplicationCorrelation and deletes one of the TestApp SNMP traps so that at most two traps are stored in the NNMi database.



Finally, after no new duplicate incidents are generated within a period of three minutes, NNMi closes the Deduplication and generates a new Deduplication incident when a new TestApp SNMP trap arrives.

Tip: The longer the time period, the more Deduplication NNMi can track.

Rate

Rate configuration enables you to track incident patterns based on the number of incident reoccurrences within a specified time period. After the count within the specified time period is reached, NNMi emits a Rate Correlation incident and continues to update the Correlation Notes with the number of occurrences within that rate. NNMi correlates the incidents under the Rate Correlation incident while they are within the specified time period. Unlike Deduplication, Rate Correlation never deletes incidents from the database.

This example configures the Rate so that NNMi generates a Rate incident when three or more TestApp SNMP traps occur within a two-minute time period.

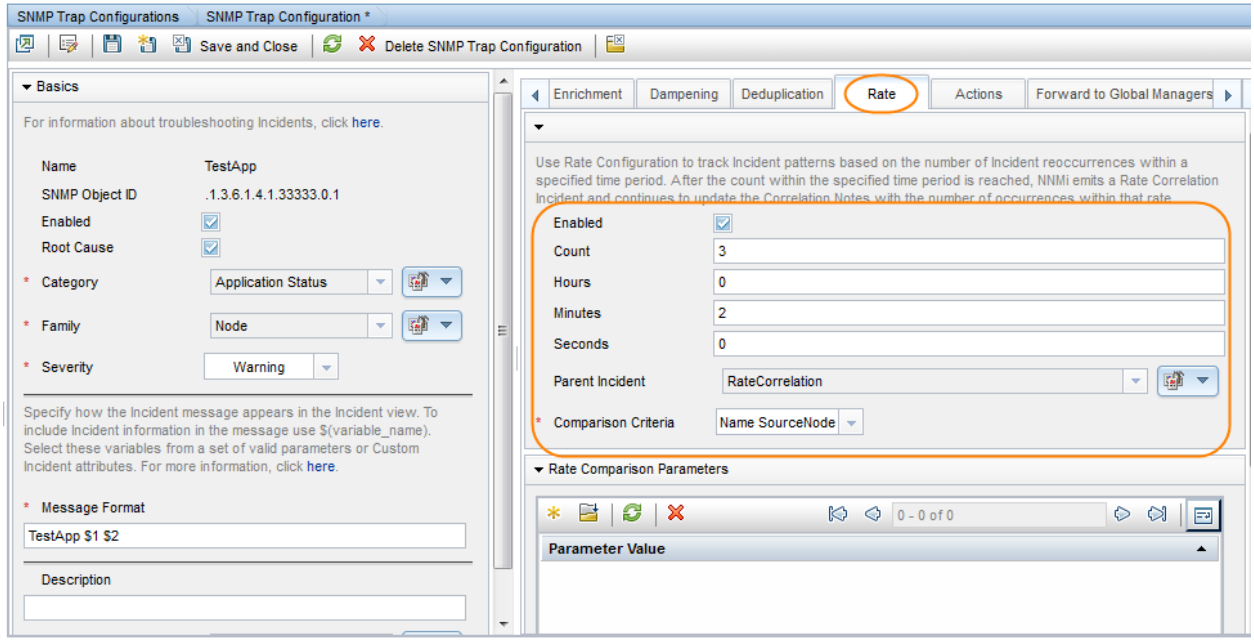
First, disable the Deduplication Incident Configuration.

1. Navigate to the **Configuration** Workspace.
2. Open **Incidents** and select **SNMP Trap Configurations**.
3. Locate and select the TestApp trap, and then click the Open button.
4. Navigate to the **Deduplication** tab.
5. Click to clear **Enabled**.

Next, specify the Rate configuration for the TestApp SNMP trap incident.

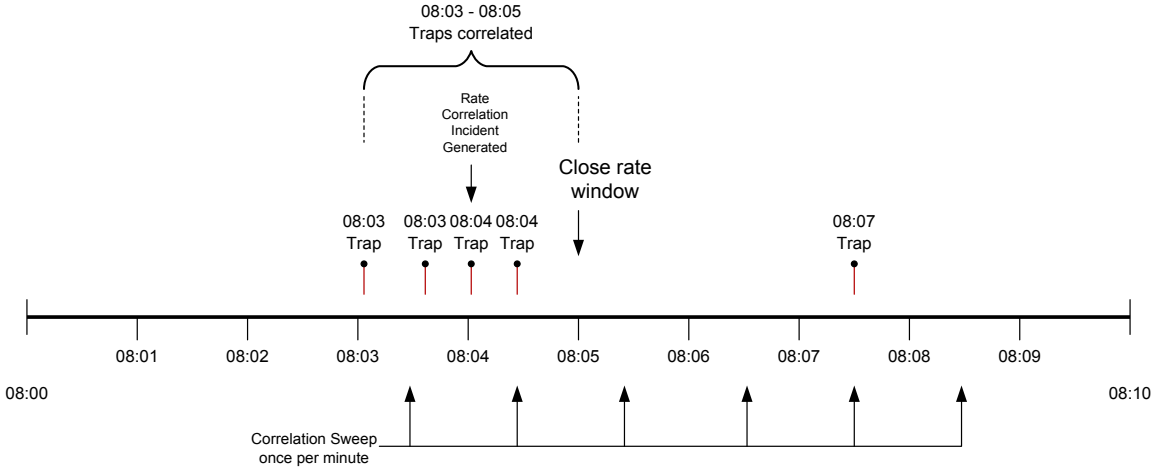
1. Navigate to the **Rate** tab.
2. Click to check **Enabled**.

3. In the Count attribute, enter **3**.
4. In the Minutes attribute, enter **2**.
5. In the **Parent Incident** drop-down list, select **RateCorrelation**.
6. In the **Comparison Criteria** drop-down list, select **Name SourceNode**.
7. Click **Save and Close** to save the configuration.



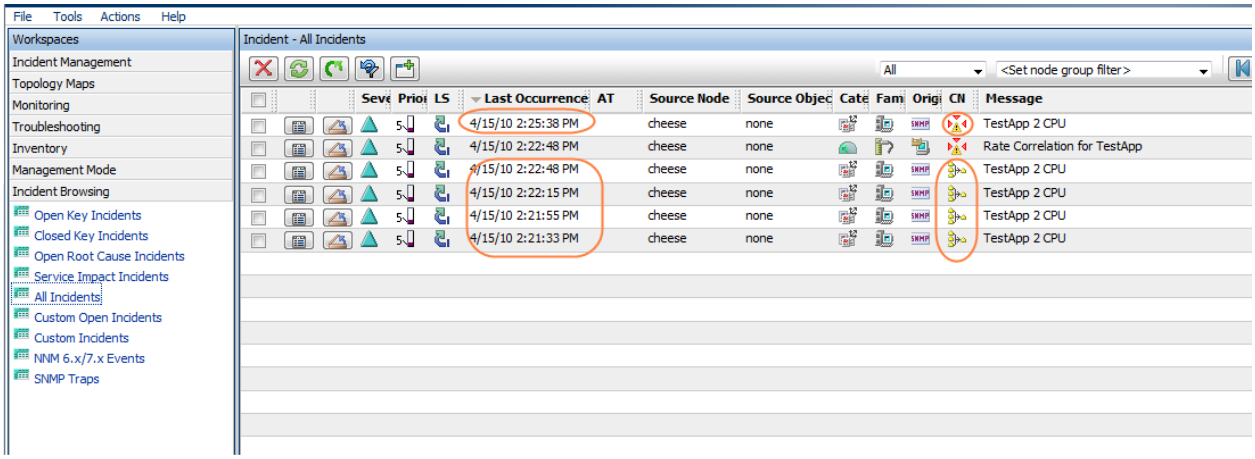
The following diagram depicts the following scenario:

The same network device generates four TestApp SNMP traps, each about 20 seconds from the previous one. Because these traps fit within the two-minute time window, when three TestApp SNMP traps occur within a two-minute time period, NNMi generates a new RateCorrelation incident. During the two-minute period, NNMi correlates all TestApp SNMP traps from this same source. After two minutes, NNMi closes the Rate time period. When another TestApp SNMP trap arrives outside of this time period, NNMi does not correlate the incident as part of this Rate correlation.

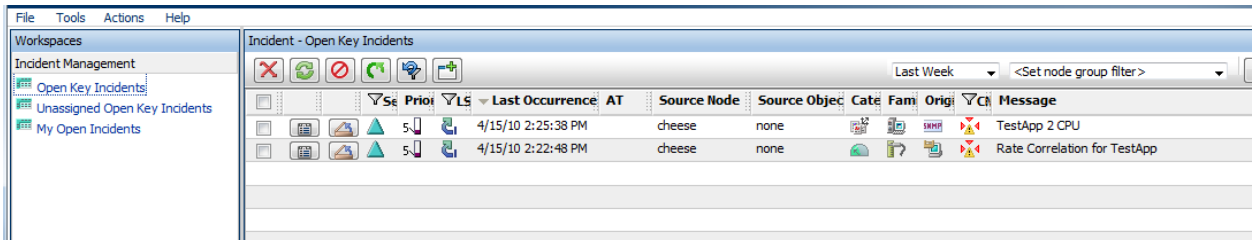


Similar to Deduplication, NNMi only checks for Rate correlations once per minute. Eventually the incident views show all the TestApp SNMP traps within the time period specified and they are correlated under the RateCorrelation incident.

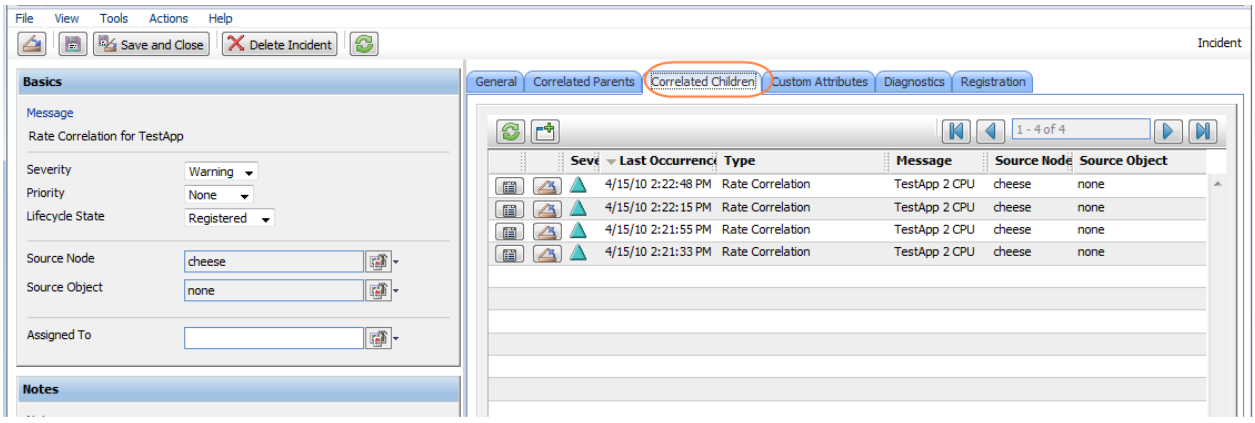
The following image depicts how these incidents might appear in the All Incidents view. As shown in the following example, NNMi correlates the first four traps under the Rate Correlation incident because they are within the two-minute window. When another TestApp SNMP trap arrives outside of the two-minute window, NNMi does not correlate the incident.



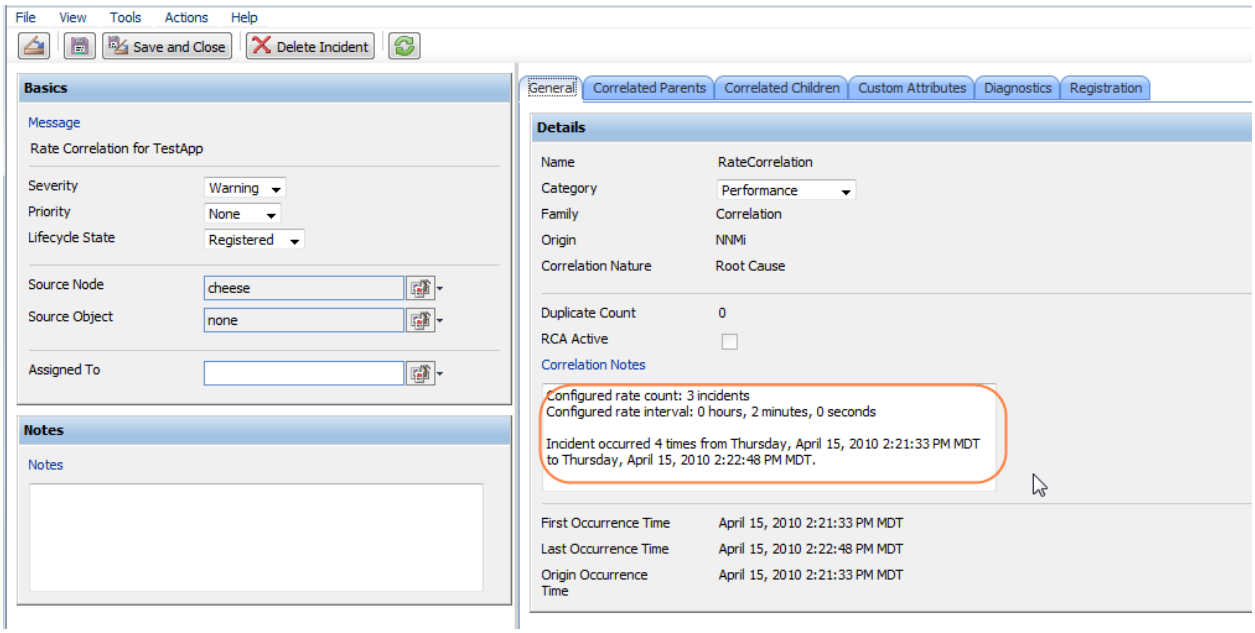
As shown in the following example, because they are not Root Cause incidents, these correlated incidents do not appear in the Open Key Incidents view.



To view the Correlated Children, open the Rate Correlation incident and navigate to the **Correlated Children** tab.



Navigate to the **General** tab to see information about the rate correlation in the Correlation Notes. The Correlation Notes are updated throughout the Rate time period that is specified.



Enrichment

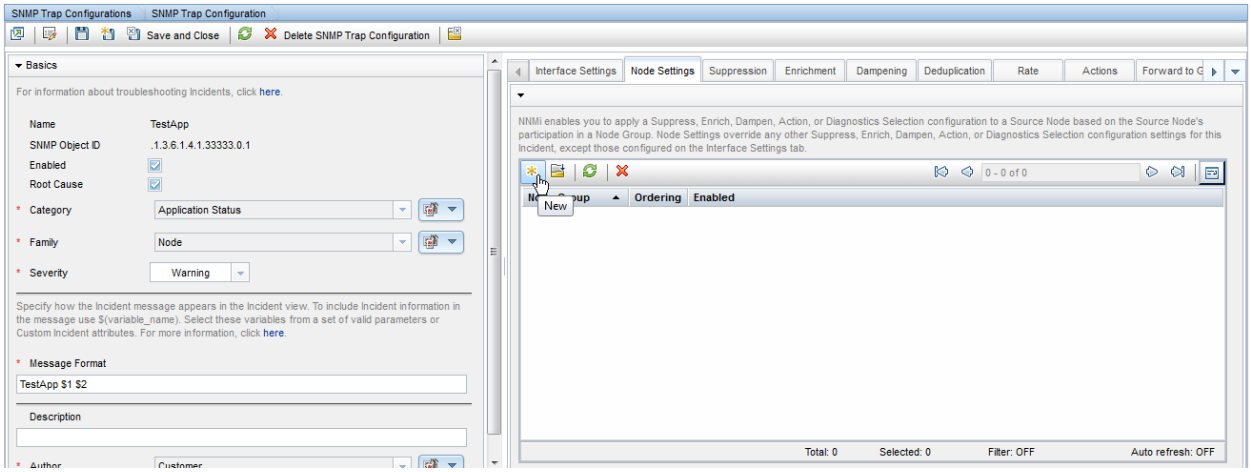
This example uses Node Group Settings with Enrichment. The Node Groups used for the Node Group Settings are named Core Routers and Important Servers.
The Enrichment feature enables you to modify an incident when it is processed by NNMi. The types of items that you can modify for a selected incident configuration include:

- Category
- Family
- Severity
- Priority
- Correlation Nature
- Message
- Assigned To
- Custom Attributes

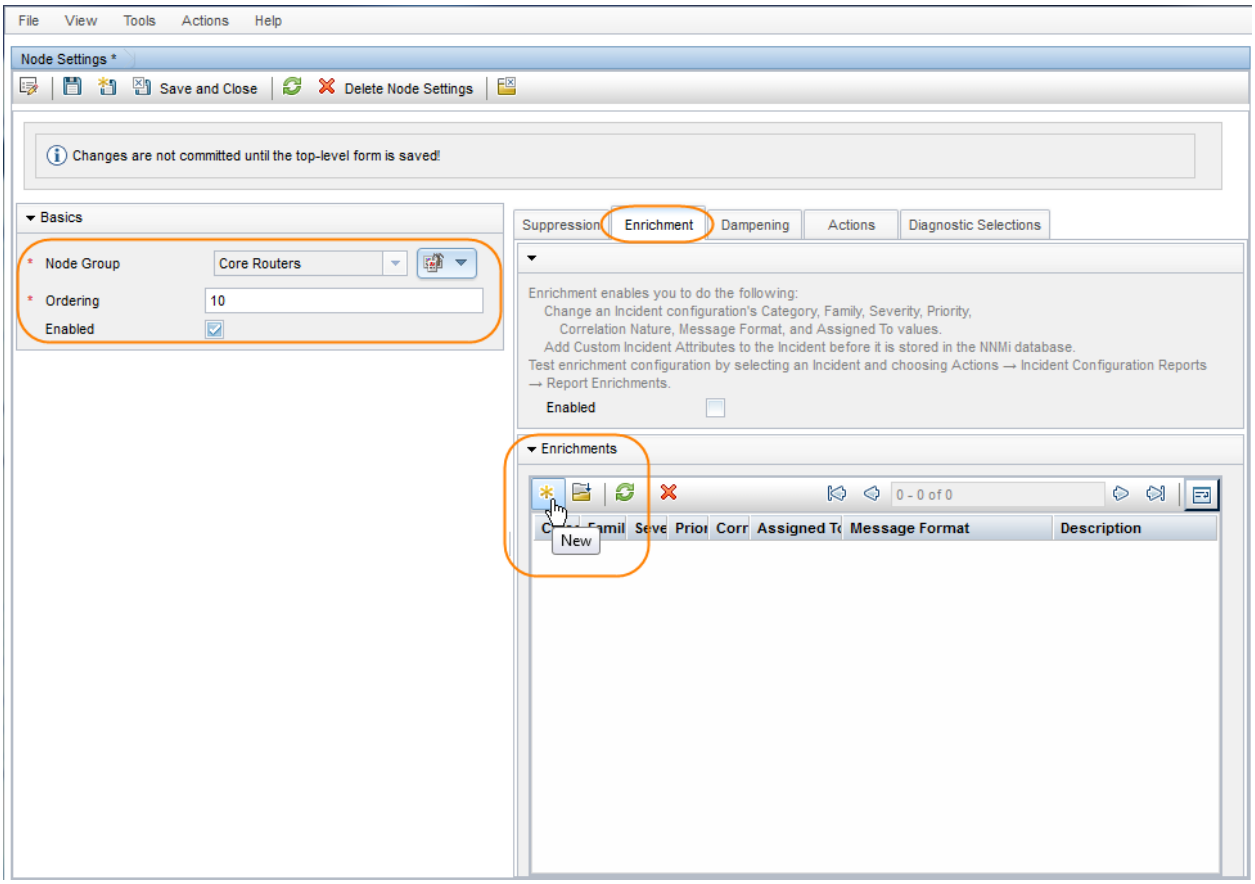
In this example, when the TestApp trap arrives from a router in the Core Routers Node Group, the Incident is enriched so that the Priority is Top. The Message Format is also customized and the incident is assigned to the user (TJ) who is in charge of the Core Routers. When the TestApp trap arrives from a server in the Important Servers Node Group, the incident is enriched so that the Priority is High. The Message Format of the incident for this Node Group is also customized.

To edit the Enrichment configuration for the TestApp SNMP trap incident:

1. Navigate to the **Configuration** Workspace.
2. Open **Incidents** and select **SNMP Trap Configurations**.
3. Locate and select the TestApp trap, and then click the Open button.
4. Navigate to the **Node Settings** tab.
5. Click the New icon.

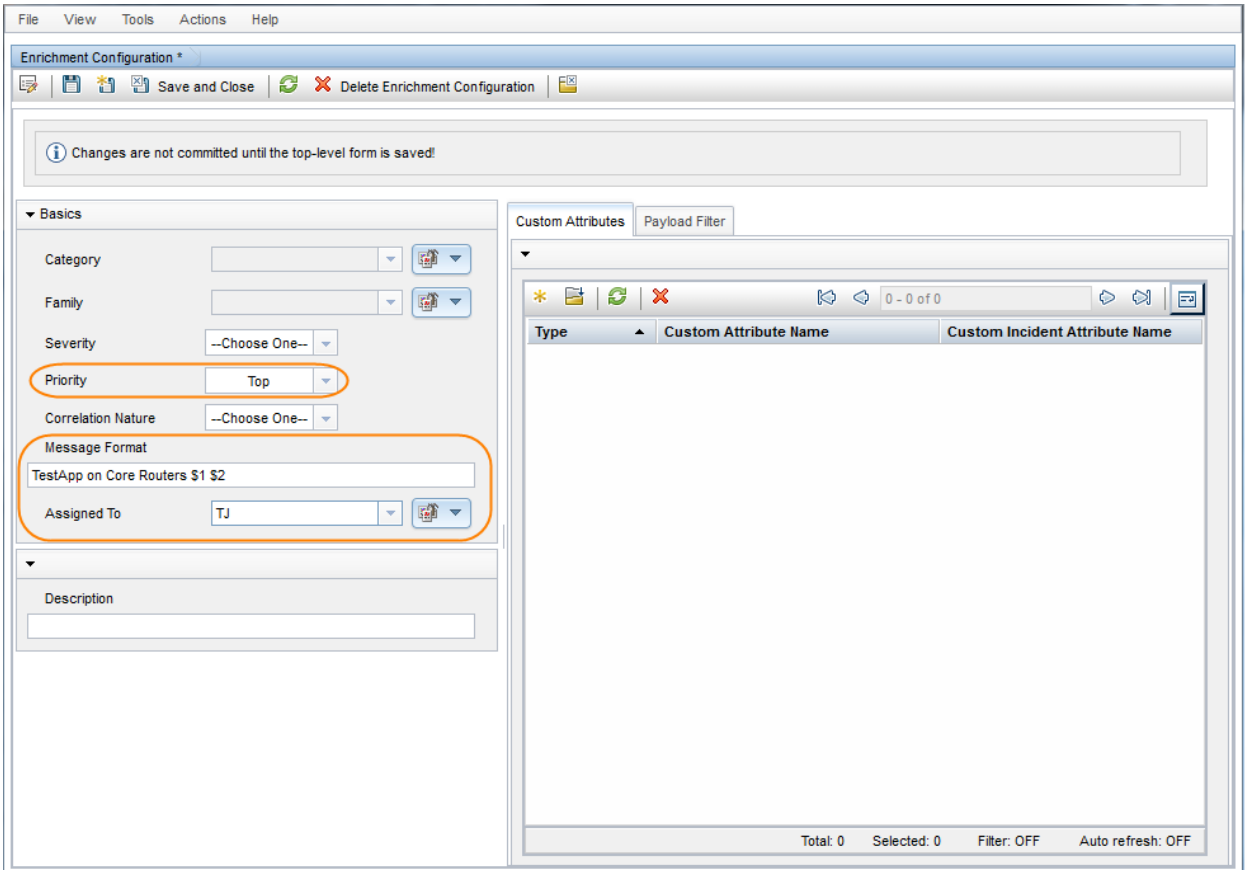


6. In the **Node Group** drop-down list, select the **Core Routers** Node Group.
7. In the **Ordering** attribute, enter **10**.
   Note: The Ordering attribute determines which Node Settings are applied to a node that is a member of more than one Node Group.
8. Click to check **Enabled**.
9. Navigate to the **Enrichment** tab.
10. Click the New * icon.



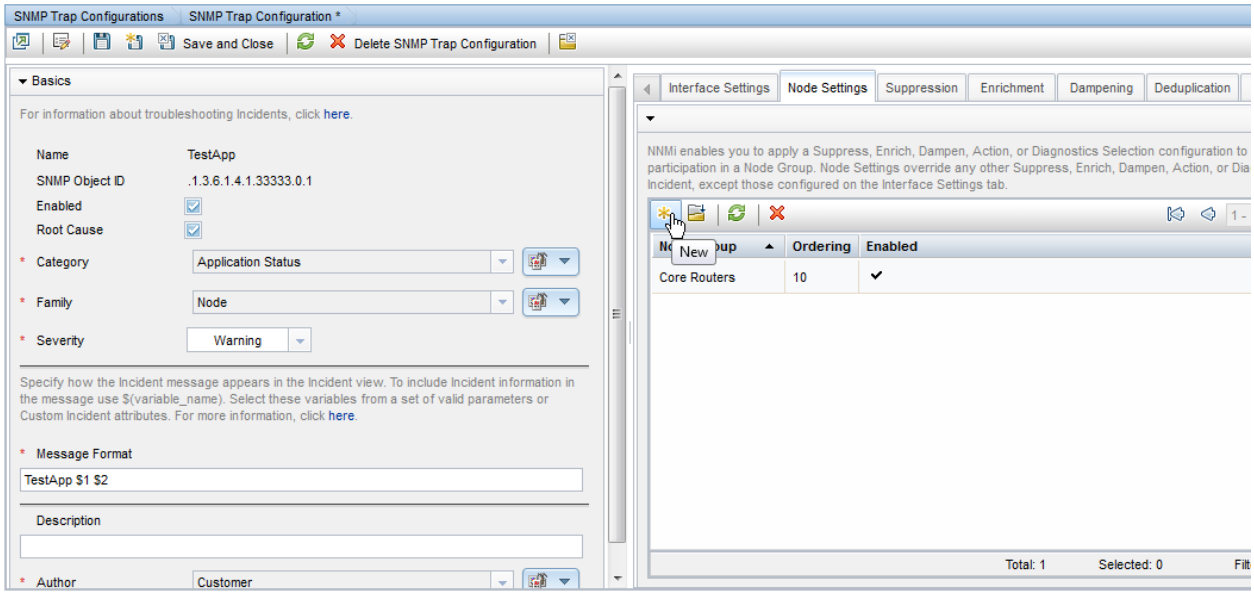
11. In the **Priority** drop-down list, select **Top**.

12. In the **Message Format** attribute, enter **TestApp on Core Routers $1 $2**.
13. In the **Assigned To** attribute, select Quick Find to select a user from the list. In this example, **TJ** is a valid user.
14. Click **Save and Close** to return to the SNMP Trap Configuration Form.
15. Click **Save and Close** to save your changes.

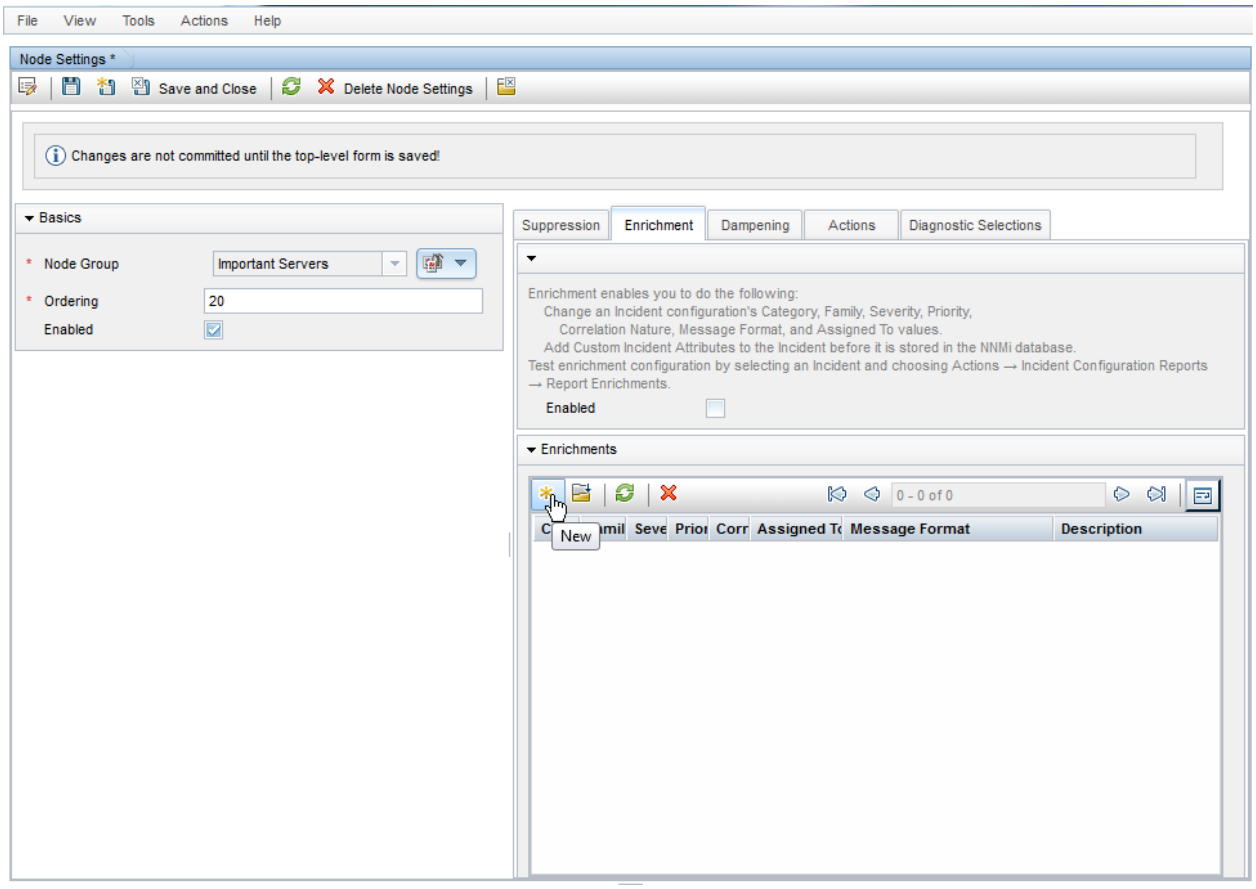


Next, configure Node Settings for the Important Servers Node Group.

1. Navigate to the **Configuration** Workspace.
2. Open **Incidents** and select **SNMP Trap Configurations**.
3. Locate and select the TestApp trap, and then click the Open button.
4. Navigate to the **Node Settings** tab.
5. Click the New icon.

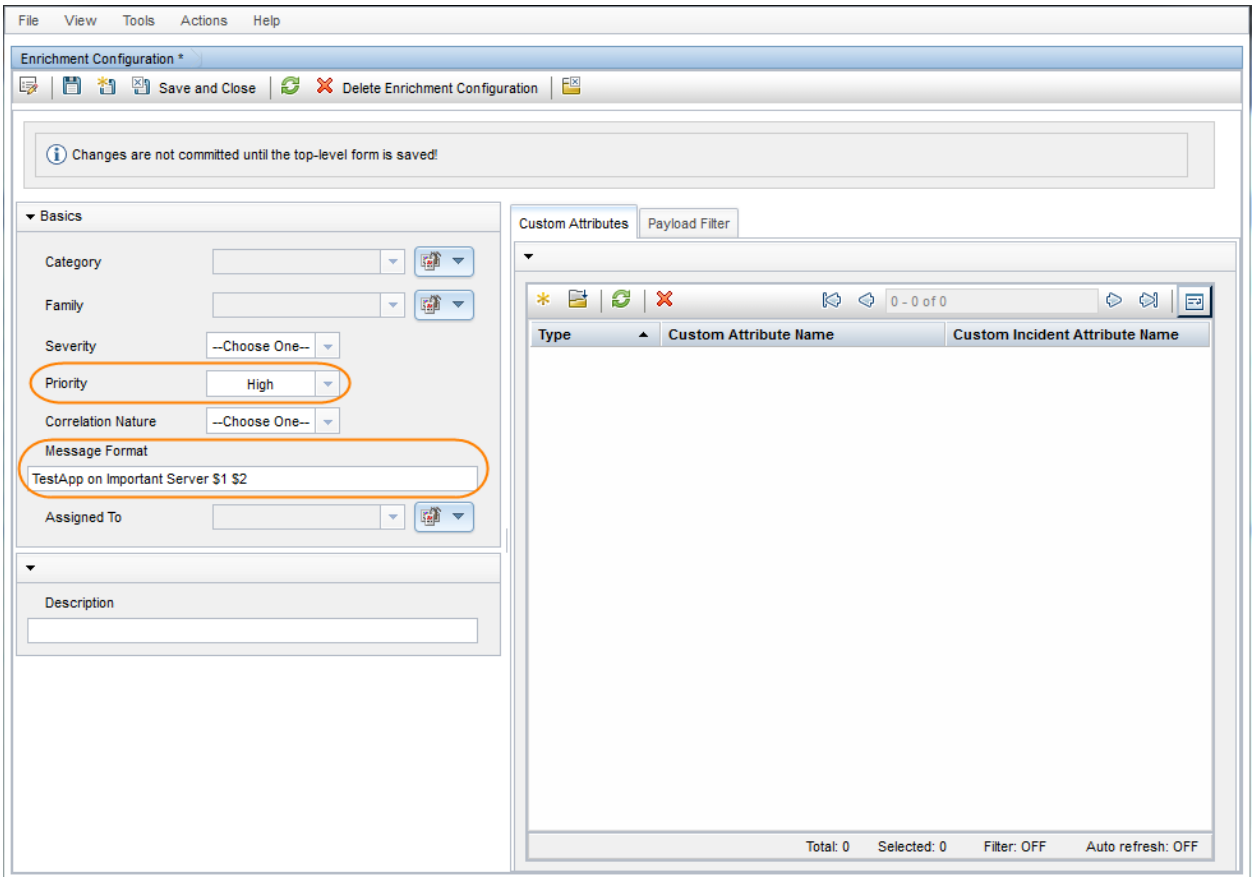


6. In the **Node Group** drop-down list, select the **Important Servers** Node Group.
7. In the **Ordering** attribute, enter **20**.
8. Click to check **Enabled**.
9. Navigate to the **Enrichment** tab.
10. Click the New * icon.

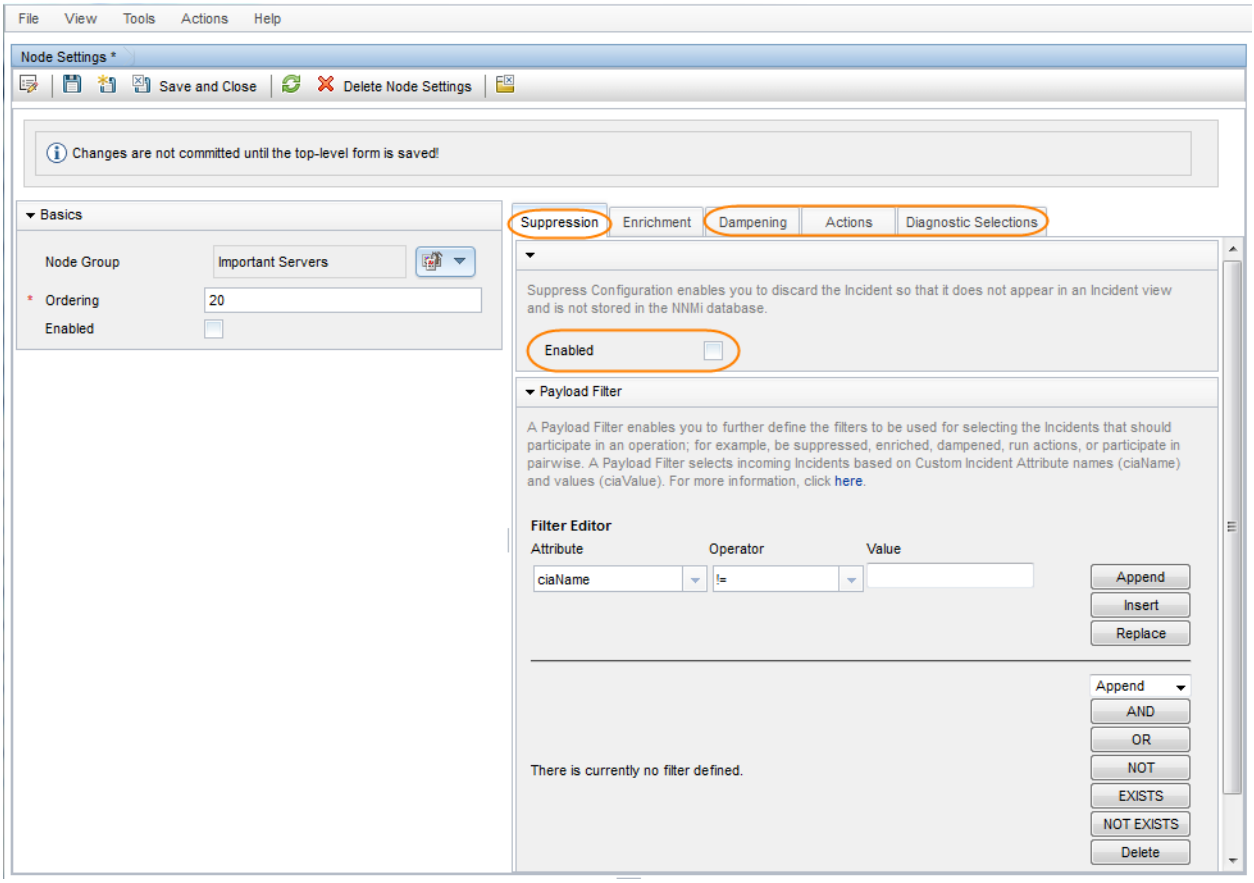


Next, set the priority to High and change the Message Format.

11. In the **Priority** attribute drop-down list, select **High**.
12. In the **Message Format** attribute, enter **TestApp on Important Server $1 $2**.
13. Click **Save and Close** to return to the SNMP Trap Configuration form.
14. Click **Save and Close** to save your changes.

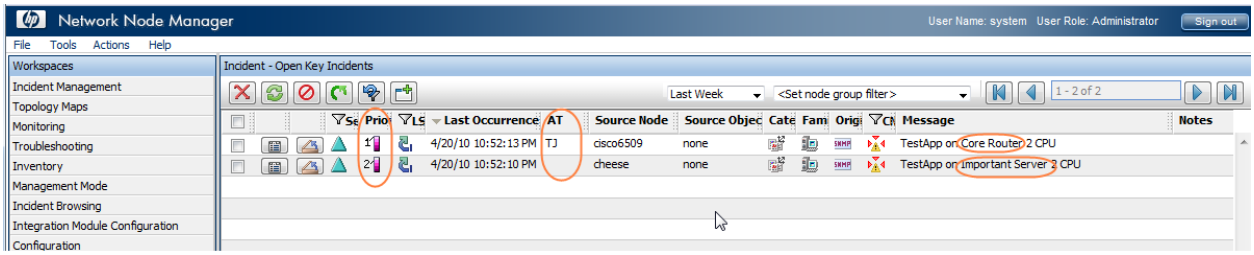


Note: When you specify Interface Settings or Node Settings, all of the Incident Configuration tabs apply for that Interface or Node Group. For example, if the Suppression configuration is not enabled, NNMi does not use the global setting for Suppression. Instead, Suppression does not occur for that incident.



Next, send a trap from each node to see the results.

As shown in the following example, the traps are enriched based on Node Group membership.

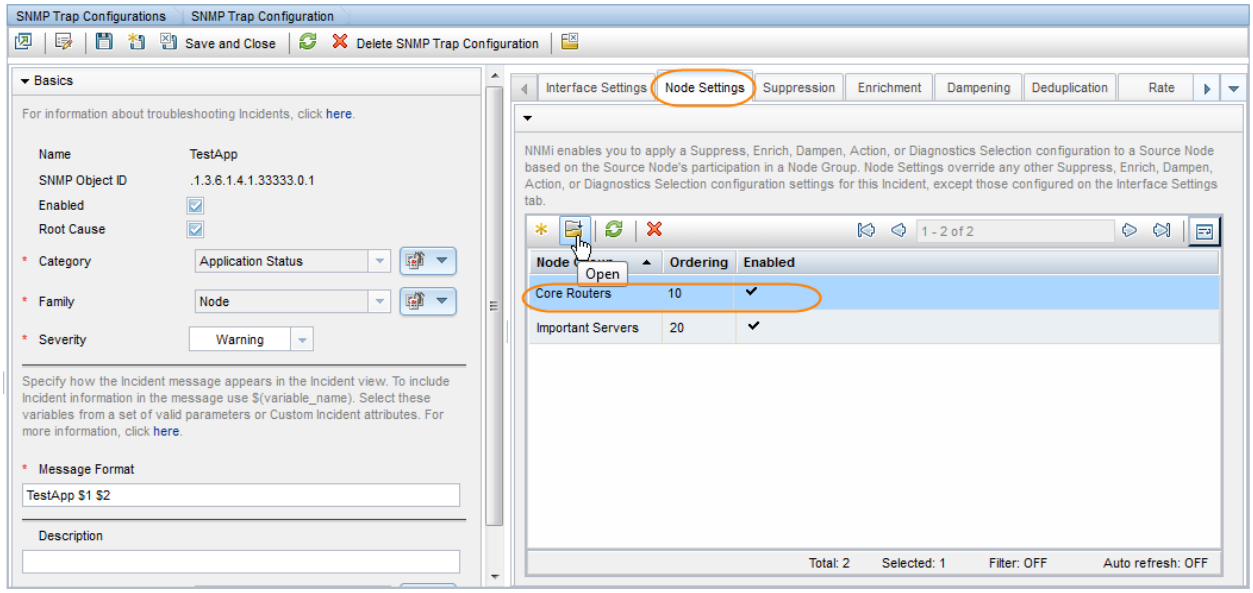


Suppression

Suppression enables you to discard traps based on specified filter values. For example, you can discard the TestApp SNMP trap incidents when the varbind value that stores Status is set to Normal or Warning for traps received from the Core Routers Node Group. This requires configuring Node Settings and Suppression.

Using the Payload Filter configuration feature, this example suppresses the trap if Varbind1=1 (Normal) or Varbind1=2 (Warning).

Tip: Use the absolute OID (Object Identifier) to specify the Varbind rather than position. For example, for Varbind1 you would specify .1.3.6.1.4.1.33333.1.1.1.



To edit the Suppression configuration for the TestApp SNMP trap incident:

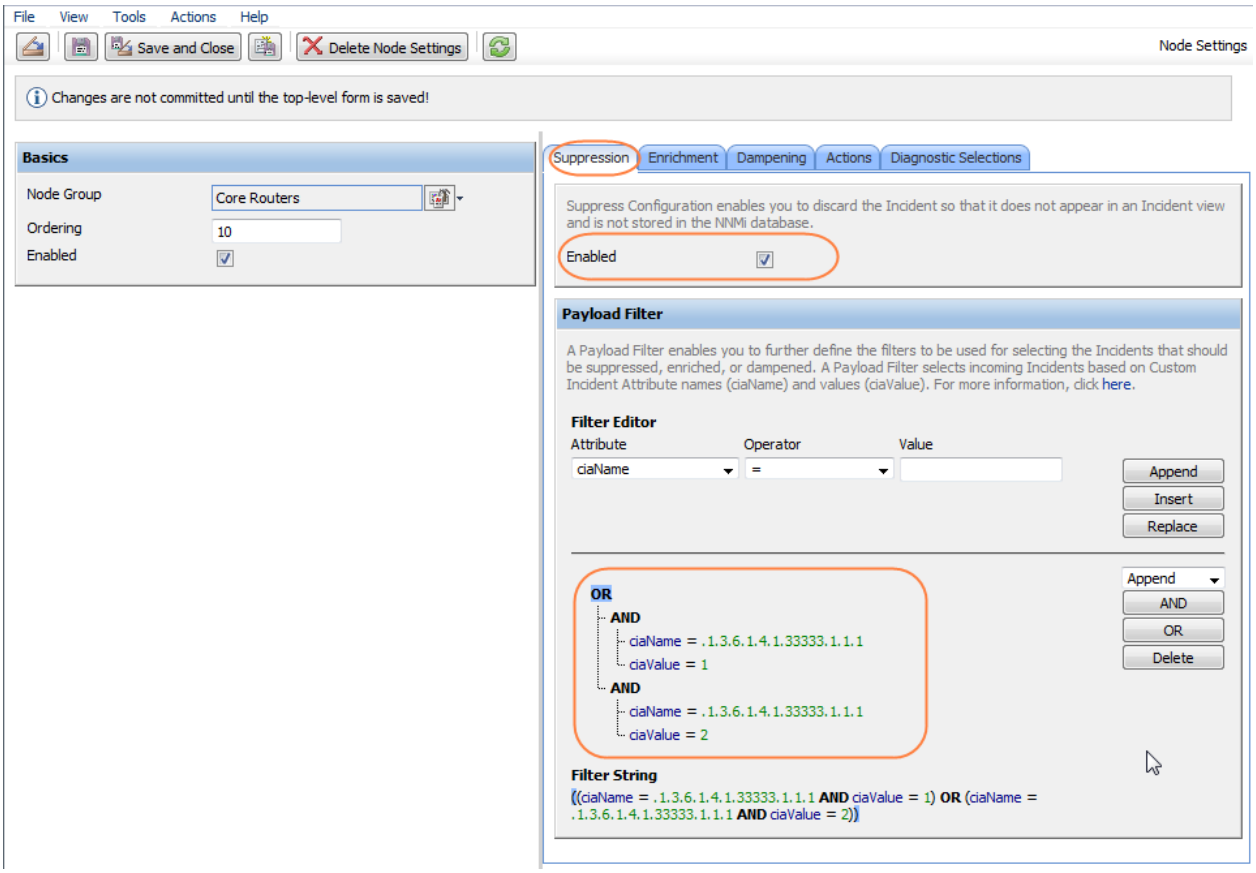
1. Navigate to the **Configuration** Workspace.
2. Open **Incidents** and select **SNMP Trap Configurations**.
3. Locate and select the TestApp trap, and then click the Open button.
4. Navigate to the **Suppression** tab.
5. Click to check **Enabled**.
6. In the Payload Filter, do the following:

Note: You must use a top level OR operator in an expression that is two levels deep as shown in this example.

   a. Make sure **Append** appears as the selection in the drop-down list.
   b. Click **OR**.
   c. Click **AND**.
   d. In the Attribute drop-down list, select **ciaName**.
   e. In the Operator attribute, select **=**.
   f. In the Value attribute, enter **.1.3.6.1.4.1.33333.1.1.1**
   g. Click **Append.**
   h. In the Attribute drop-down list, select **ciaValue**.
   i. In the Operator attribute, select **=**.
   j. In the Value attribute, enter **1**.
   k. Click **Append**.
   l. Click **AND**.
   m. In the Attribute drop-down list, select **ciaName**.
   n. In the Operator attribute, select **=**.
   o. In the Value attribute, enter **.1.3.6.1.4.1.33333.1.1.1**

p. Click **Append.**
q. In the Attribute drop-down list, select **ciaValue**.
r. In the Operator attribute, select **=**.
s. In the Value attribute, enter **2**.
t. Click **Append**.

7. Click **Save and Close** to return to the SNMP Trap Configuration form.
8. Click **Save and Close** to save your changes.

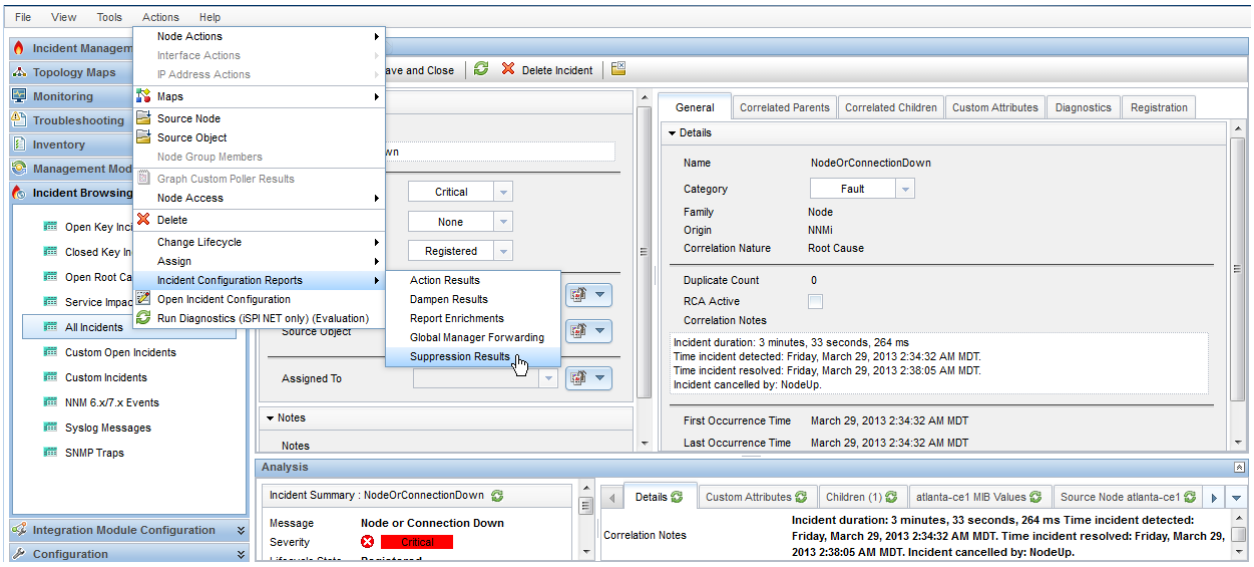


To determine whether an SNMP trap incident is being suppressed, examine one of the SNMP traps in the NNMi database that has already been sent and not suppressed:

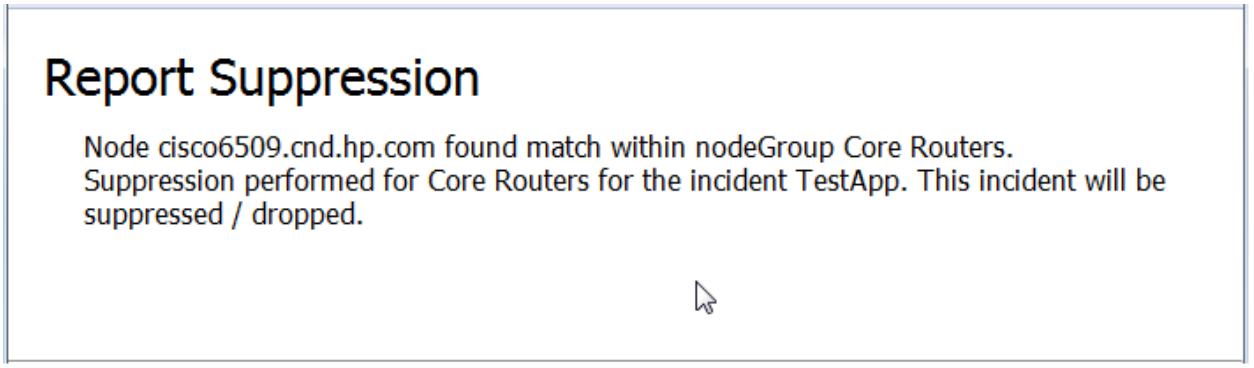
1. From an incident view, select the incident of interest and then click the Open button.
2. Select **Actions->Incident Configuration Reports->Suppression Results**.

The Suppression Results report displays the results of processing the incident using the Suppression configuration specified for that incident as if the incident was generated.

Note: The Suppression Results report does not actually execute the rules, but instead reports on how the Suppression configuration would be executed. This report is useful to determine whether the Suppression configuration matches any incidents. You can use the same approach for Actions, Dampen, and Enrichment configurations as well.



The following example verifies that a match is made and this trap would be suppressed if received.



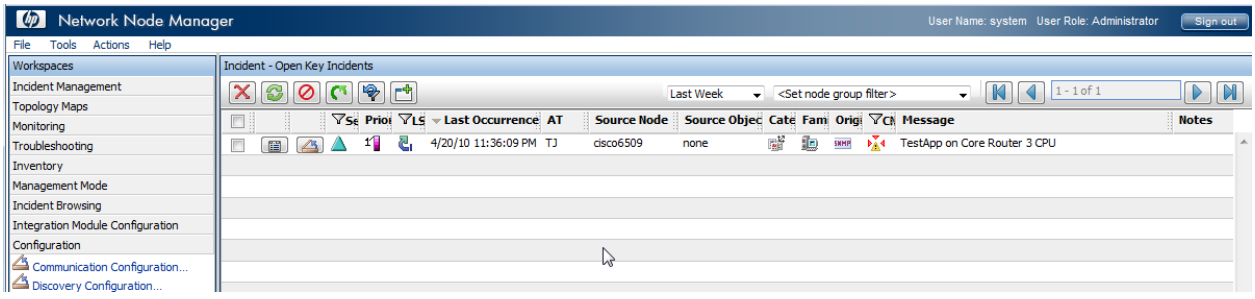
To fully test the Suppression configuration, send the trap three times, each with a different Varbind value (1, 2, and 3):

```
# nnmsnmpnotify.ovpl -a 15.6.96.97 localhost .1.3.6.1.4.1.33333.0.1 .1.3.6.1.4.1.33333.1.1.1
integer 1  .1.3.6.1.4.1.33333.1.2.1 OCTETSTRING CPU

# nnmsnmpnotify.ovpl -a 15.6.96.97 localhost .1.3.6.1.4.1.33333.0.1 .1.3.6.1.4.1.33333.1.1.1
integer 2  .1.3.6.1.4.1.33333.1.2.1 OCTETSTRING CPU

# nnmsnmpnotify.ovpl -a 15.6.96.97 localhost .1.3.6.1.4.1.33333.0.1 .1.3.6.1.4.1.33333.1.1.1
integer 3  .1.3.6.1.4.1.33333.1.2.1 OCTETSTRING CPU
```

Notice that only the TestApp SNMP trap with Varind1=3 appears in the Open Key Incidents view. NNMi suppresses the other two TestApp SNMP trap incidents.



More about Dampening

The Dampening feature is useful for incidents that NNMi closes automatically when the condition is cleared rather than the simple traps included in the previous examples. For example, NNMi closes the InterfaceDown incident when the status of the interface goes to Normal. If this were to occur during the Dampening period, NNMi does not display the incident in any Incident Management or Incident Browsing views.

By default, NNMi dampens the Management Events it provides for a period of 6 minutes. Dampening can be configured to a maximum of one hour to allow two polling cycles to occur before NNMi sets the Lifecycle State to Registered.

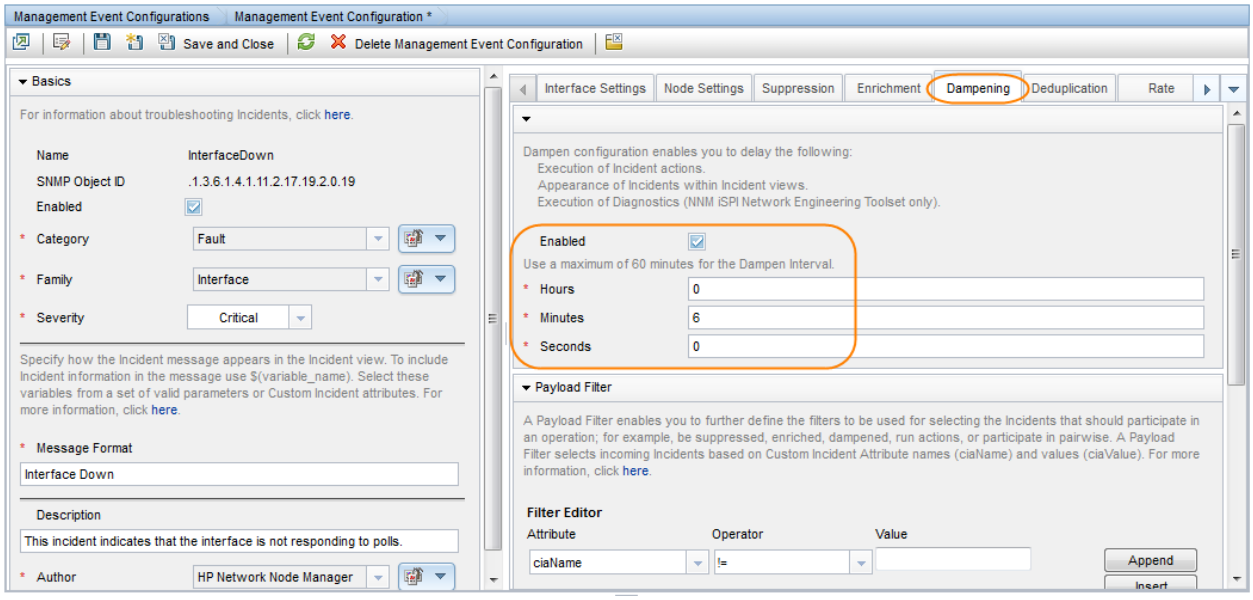
To disable the Dampening for an incident configuration, click to clear **Enabled** on the incident configuration form.

You can also use the `nnmsetdampenedinterval.ovpl` command line tool to set the Dampening period and enable Dampening for all incidents.

To disable Dampening for all incident configurations use `nnmsetdampenedinterval.ovpl` as shown in the following example:

```
nnmsetdampenedinterval.ovpl -hours 0 -minutes 0 -seconds 0
```

An example of the dampening for the InterfaceDown incident is shown below:



Lifecycle State and Actions

NNMi has four common Lifecycle States: Registered, In Progress, Completed, and Closed. It is important to understand Lifecycle State changes because these state changes are the triggers for actions in NNMi.

It is also important to understand that NNMi changes the Lifecycle State to Closed based on the "Down" incident. For example, when an interface goes down, an Interface Down incident is generated and, if the incident is not Dampened, NNMi sets the Lifecycle State to Registered. When the interface comes back up again, NNMi changes the Lifecycle State to Closed, but does not generate an additional Interface Up incident.

This example uses two command line scripts that can be run as actions. One script (ServerScript.ksh) is to be run for TestApp traps that arrive from the Important Servers group. The other script (RouterScript.ksh) is to be run on traps that arrive from the Core Routers group. Each script is passed Source Node Name ($snn) as well as the Varbind1 and Varbind2 values.

The two scripts are as follows:

ServerScript.ksh:

```
#!/usr/bin/ksh
echo $1 $2 $3 >> /tmp/serverscript.txt
```

RouterScript.ksh:

```
#!/usr/bin/ksh
echo $1 $2 $3 >> /tmp/routerscript.txt
```

1. Place the scripts into the following directory and make sure they are executable:

Windows:

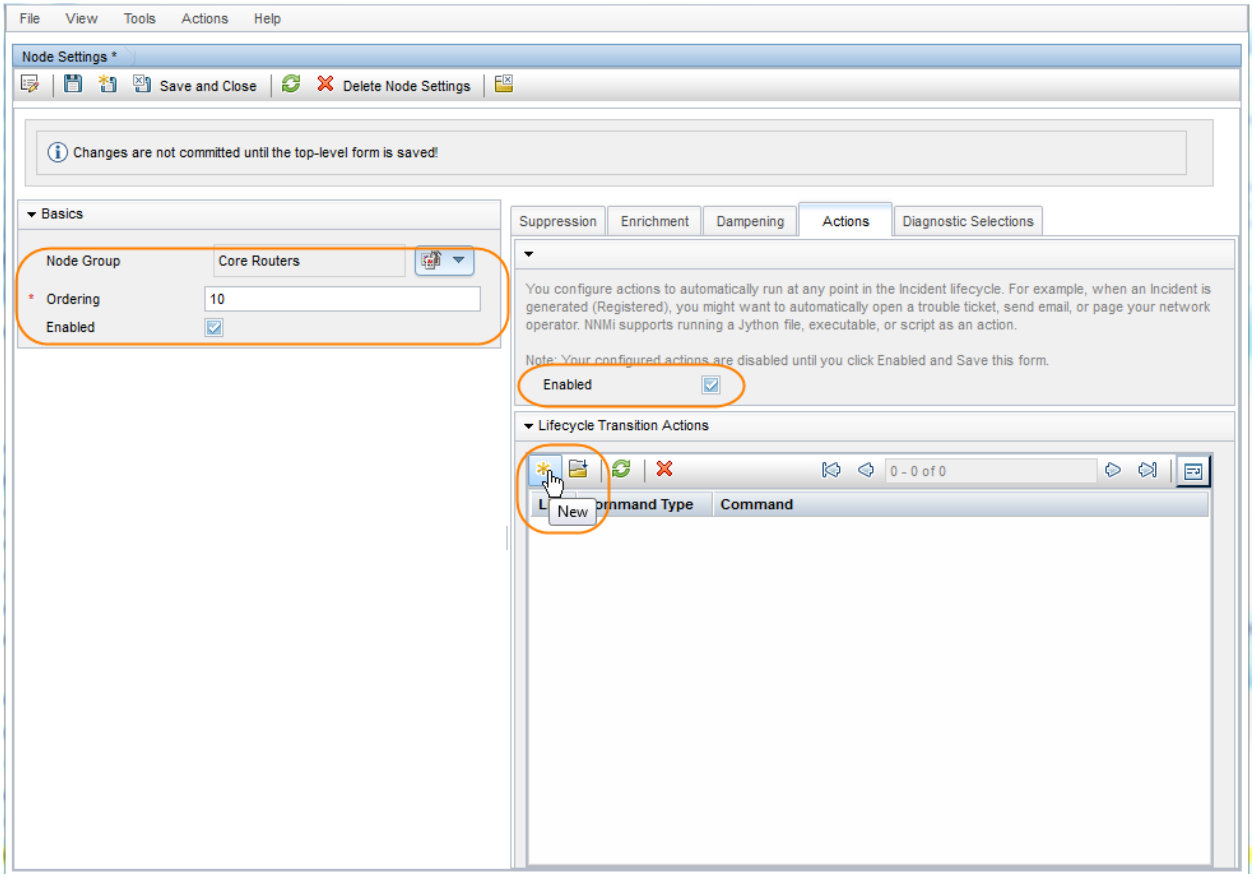
%NnmDataDir%\shared\nnm\actions

UNIX:

/var/opt/OV/shared/nnm/actions

To configure these scripts as actions for the TestApp SNMP trap incident configuration:

2. Navigate to the **Configuration** Workspace.
3. Open **Incidents** and select **SNMP Trap Configurations**.
4. Locate and select the TestApp trap, and then click the Open button.
5. Navigate to the **Node Settings** tab.
6. Select Core Routers and then click the Open button.
7. In the **Node Group** drop-down list, select the **Core Routers** Node Group.
8. In the **Ordering** attribute, enter **10**.
9. Click to check **Enabled**.
10. Navigate to the **Actions** tab.
11. Click to check **Enabled**.
12. Click the New icon.



Next, specify the action to be run and the arguments to pass it.

You can specify Varbinds, using the full OID (as shown below) or using a position number, such as $1 and $2. The advantage to using the full OID is that the action can be re-run on an "already received trap". NNMi does not store the Varbind position, but if you use the OID specification, it properly re-runs the action as demonstrated in this example.

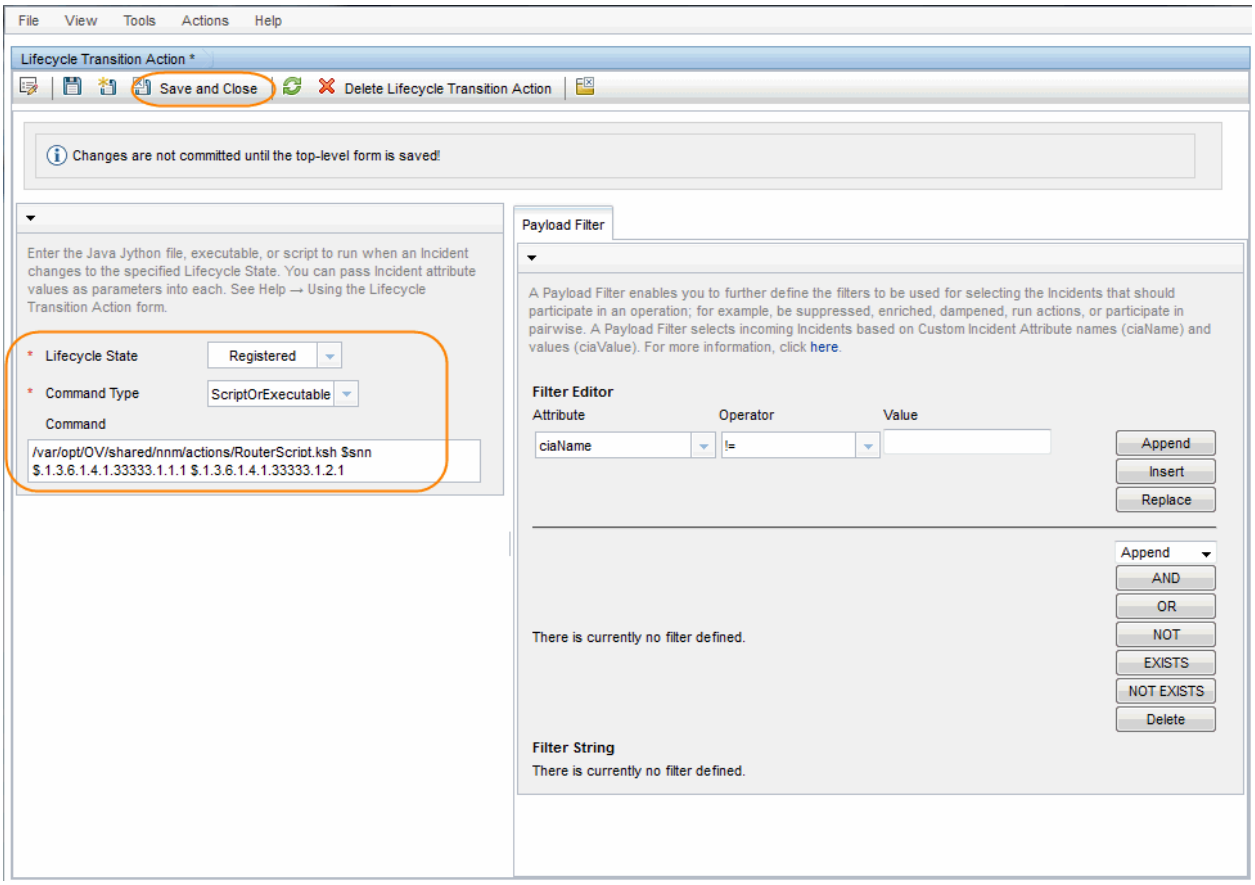
To configure a Lifecycle Transition Action:

1. In the **Lifecycle State** drop-down list, select **Registered**.
2. In the **Command Type** drop-down list, select **ScriptOrExecutable**.
3. In the **Command** attribute, enter the following command:

```
/var/opt/OV/shared/nnm/actions/RouterScript.ksh $snn $.1.3.6.1.4.1.33333.1.1.1
$.1.3.6.1.4.1.33333.1.2.1
```

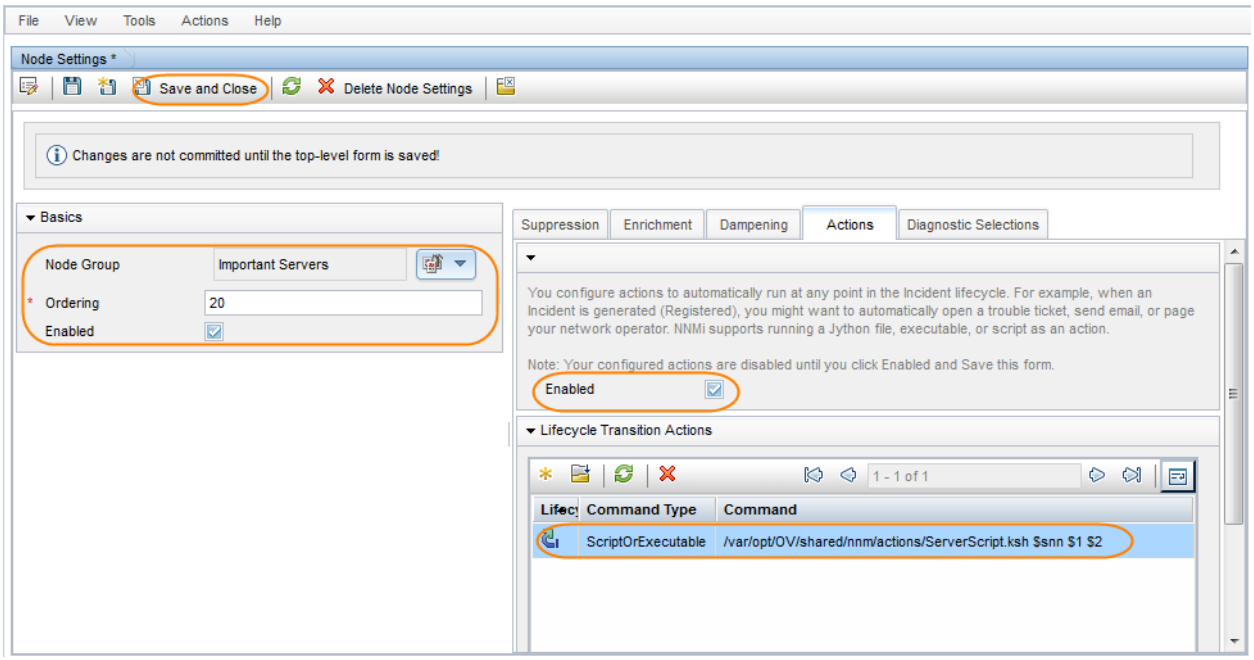
Tip: Include the full path to the action script.

4. Click **Save and Close** to save your changes.



Next configure the action for the Important Servers Node Group.

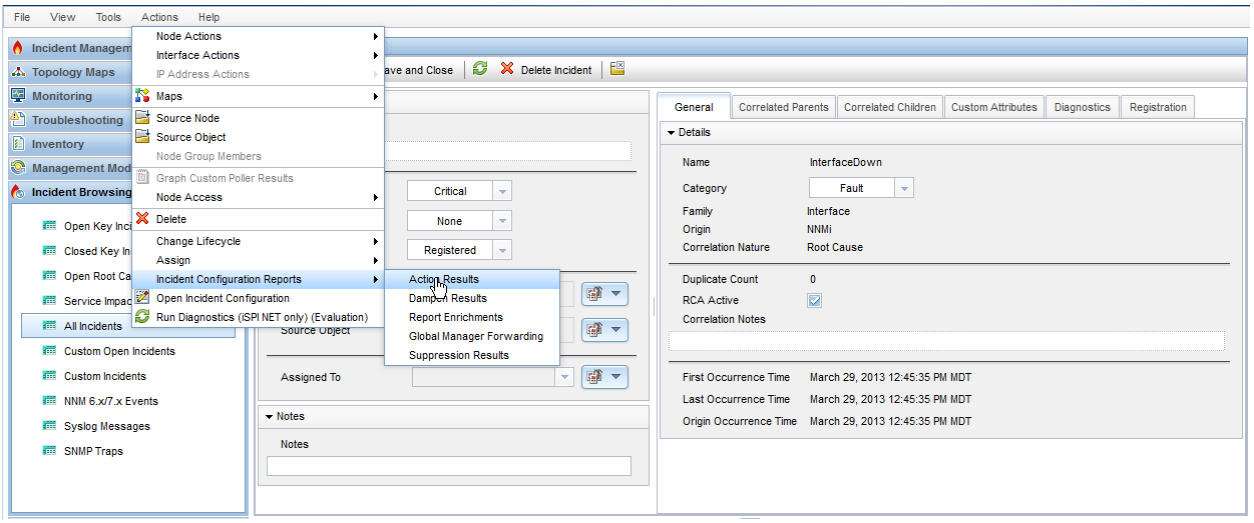
1. Navigate to the **Configuration** Workspace.
2. Open **Incidents** and select **SNMP Trap Configurations**.
3. Locate and select the TestApp trap, and then click the Open button.
4. Navigate to the **Node Settings** tab.
5. In the **Node Group** list, select the **Important Servers** Node Group and click the Open button.
6. In the **Ordering** attribute, enter **20**.
7. Click to check **Enabled**.
8. Navigate to the **Actions** tab.
9. Click the New icon.
10. Click to check **Enabled**.
11. In the **Lifecycle State** drop-down list, select **Registered**.
12. In the **Command Type** drop-down list, select **ScriptOrExecutable**.
13. In the **Command** attribute, enter the following command:
    `/var/opt/OV/shared/nnm/actions/ServerScript.ksh $snn $1 $2`
14. Click **Save and Close**.



15. Click **Save and Close** to save your changes.
16. Click **Save and Close** again to save your changes.

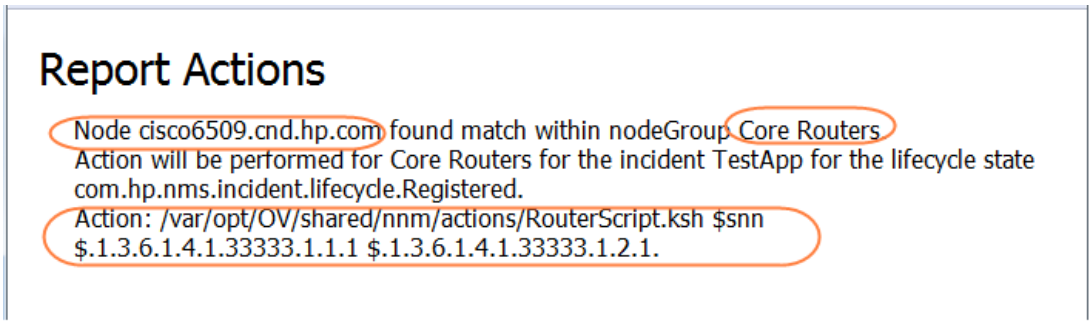
To confirm that action is configured properly:

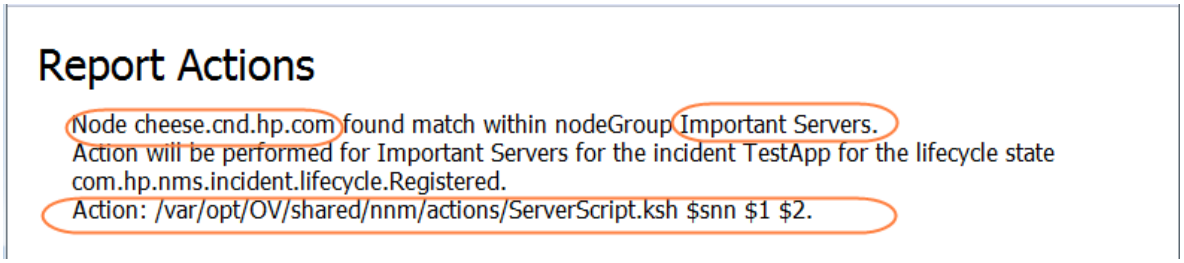
1. From an incident view, select an incident of interest and then click the Open button.
2. Select **Actions->Incident Configuration Reports-> Action Results**.



The Action Results report displays whether a Node Group match occurred for that particular trap and if the action would have been run.

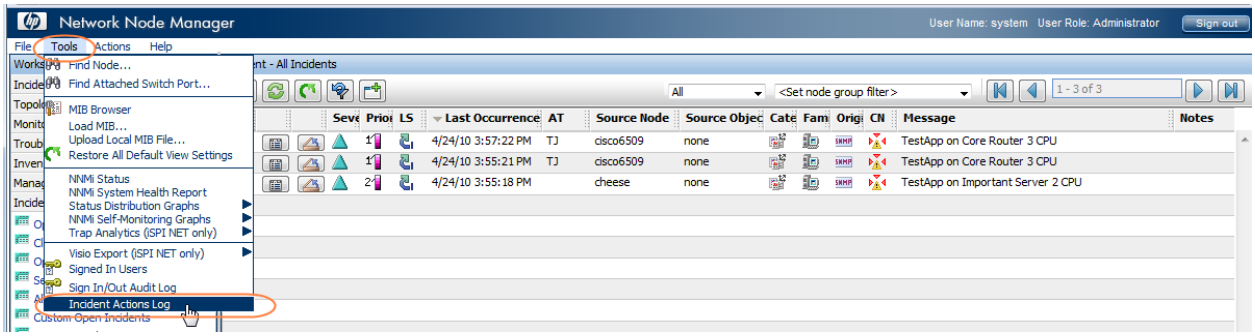
Note: Run the Action Results report for a node in the Core Routers group and for a node in the Important Servers group.





Next, send one of the traps.

After the trap is sent, check the Incident Actions log for a message indicating the action was run.



You can also check the results of the action as shown in the following example:

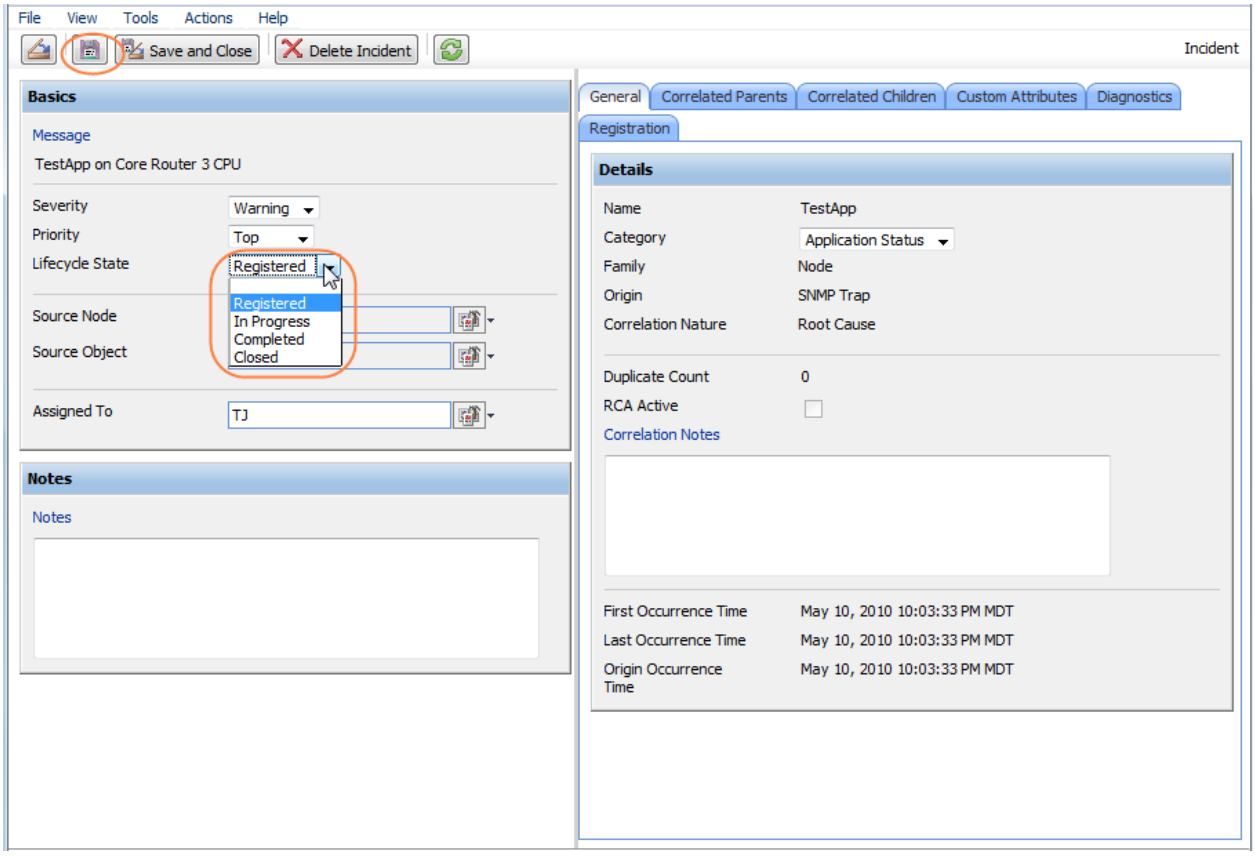
```
# cat /tmp/serverscript.txt
cheese 2 CPU
```

To practice running the action from an already received incident:

1. From an incident view, select an incident of interest and then click the Open button.
2. Change the Lifecycle State attribute to a different state.
3. Click **Save and Close**.

4. Change the Lifecycle State attribute value back to the Registered State.
5. Click **Save and Close**.

Note: NNMi processes the varbinds values in the proper order when the trap first arrives, but it does not do so in subsequent runs when the Lifecycle State is changed and the varbinds are identified using position number. Therefore, use the full OIDs for the varbinds when forcing a Lifecycle State change.



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

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

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 and associated patch documentation
- 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