# Step-by-Step Guide to Managing SNMP Traps in NNMi

Managing SNMP trap configuration within NNMi is a key task that can be used to improve NNMi efficiency. This document steps through the following three scenarios for configuring and managing traps with NNMi 8.10:

- Determine why SNMP traps are not appearing in the NNMi Console
- Manage the Number of Incoming SNMP Traps
- Periodically trim the NNMi database to remove SNMP traps

When following the procedures in this document, note the following:

- The examples in this document use a UNIX NNMi management server, but can be followed on a Windows NNMi management server.
- The example commands assume the system user is assigned the following password: `mypassword`
- Some screen captures might be slightly different from those that appear in the most recent NNMi graphical user interface.

This document begins with some background information about NNMi and SNMP traps.

## About NNMi and SNMP Traps

NNMi works with SNMP traps at the following two levels:

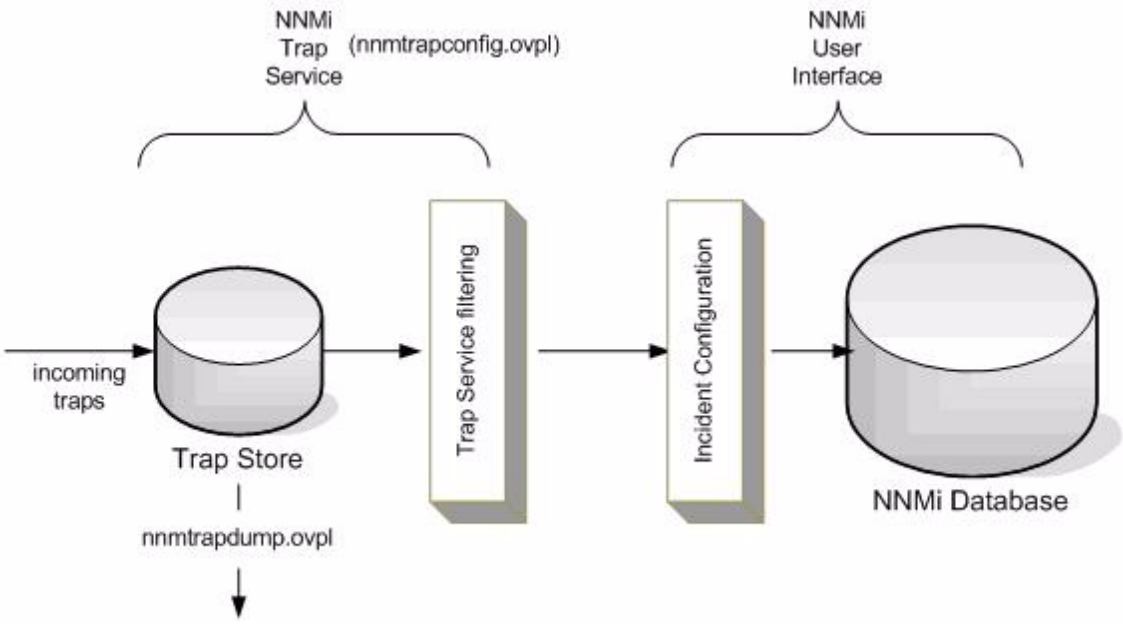
- NNMi Trap Service
- Event Subsystem

The NNMi Trap Service maintains a separate data store of traps. This service is managed through the `nnmtrapconfig.ovpl` tool.

The event subsystem is more visible and is controlled through the NNMi user interface.

As shown in the following diagram, incoming traps first pass through the NNMi Trap Service.

**Figure 1 Understanding the Flow of SNMP Traps**




Understanding the flow of SNMP traps within NNMi will help you troubleshoot unexpected SNMP trap behavior as described in the next section.

# Task 1: Determine why SNMP Traps are not Appearing in the NNMi Console

This example steps through troubleshooting the following scenario:

SNMP traps are being sent to the NNMi management server, but are not displayed in the NNMi SNMP Trap views.

This example uses the following trap and node values:

- SNMP Trap: `ciscoSyslogMIBNotification trap (.1.3.6.1.4.1.9.9.41.2.0.1)`
- Node: `10.2.121.254`

To determine why SNMP traps are not appearing in the NNMi console:

1. Check that the trap service is up and running, using the `ovstatus` command:

   **`ovstatus -v ovjboss`**

   The output should include the following line to indicate the SNMP Trap Service is running:

   ```
   NnmTrapService                          Service is started
   ```

2. Check the SNMP Trap Service configuration using `nnmtrapconfig.ovpl`:

   **`nnmtrapconfig.ovpl -showProp -u system -p mypassword`**

Compare your defaults with the default values in the following example `nnmtrapconfig.ovpl` output:

Many parameters can be tuned by using the `nnmtrapconfig.ovpl tool`. See the `nnmtrapconfig.ovpl` Reference Page (`Help --> Documentation Library --> Reference Pages`) for more information.

```
trapInterface        : All interfaces
trapPort             : 162
recvSocketBufSize    : 2048 KBytes
blockTraps           : true
thresholdRate        : 50 traps/sec
rearmRate            : 50 traps/sec
overallThresholdRate : 150 traps/sec
overallRearmRate     : 150 traps/sec
windowSize           : 300 secs
updateSourcesPeriod  : 30 secs
notifySourcesPeriod  : 300 secs
minTrapCount         : 25 traps
numSources           : 10
databaseQSize        : 300000 traps
pipelineQSize        : 50000 traps
databaseFileSize     : 100 MBytes
databaseFileCount    : 5
loopbackAddrOverride : Empty
```

3 After you know that the service is running and configured, check the NNMi Trap Service storage (Trap Store) files to determine whether they are getting larger.

Unix Example

a Navigate to the following directory:

**`/var/opt/OV/shared/nnm/databases/traps`**

b List the directory contents. For example, on UNIX:

**`ls -l`**

You should see a maximum of five files: `traplog0` through `traplog4`. You can view the date stamps to see how fast the log files are being updated.

4 Use the `nnmtrapdump.ovpl` command to verify that the traps you are expecting to view in NNMi appear in the contents of the NNMi Trap Service storage (Trap Store).

It is recommended that you redirect the output. Because the output can be quite large, you might want to limit the amount of traps in the output using any of the following options: `-source`, `-trapid`, `-from`, and `-to`.

For our example, we use the `-source` option with a value of 10.2.121.254 to limit the amount of traps. coming from the source device that has an IP address of 10.2.121.254.

```
nnmtrapdump.ovpl -u system -p mypassword -source\
10.2.121.254 > /tmp/trapdump.txt
```

Your output should appear similar to the following example `nnmtrapdump.ovpl` output. As shown in the example output, each entry begins with the Trap ID and then gives additional information

```
Trap .1.3.6.1.4.1.9.9.41.2.0.1 at October 31, 2008 3:20:30 PM
MDT from cisco6509.cnd.hp.com

Version: SNMPv1

Enterprise OID: .1.3.6.1.4.1.9.9.41.2

Agent address: 10.2.121.254

Generic trap: 6

Specific trap: 1

Timeticks: 1,358,199,695

Varbinds:

<VB state='0' type='4'
id='.1.3.6.1.4.1.9.9.41.1.2.3.1.2.2865238' value='SNMP' />

<VB state='0' type='2'
id='.1.3.6.1.4.1.9.9.41.1.2.3.1.3.2865238' value='4' />

<VB state='0' type='4'
id='.1.3.6.1.4.1.9.9.41.1.2.3.1.4.2865238' value='AUTHFAIL' />

<VB state='0' type='4'
id='.1.3.6.1.4.1.9.9.41.1.2.3.1.5.2865238'
value='Authentication failure for SNMP req from host
16.157.128.85' />

<VB state='0' type='67'
id='.1.3.6.1.4.1.9.9.41.1.2.3.1.6.2865238'
value='1358199695' />
```

If you find the traps you are expecting in the output, this verifies that the traps are arriving into the NNMi Trap Service storage (Trap Store).

If you verify that an SNMP trap is arriving into the NNMi Trap Service storage (Trap Store), but you cannot see the SNMP trap in the NNMi console, the SNMP trap must be filtered after it has been received by the NNMi Trap Service.

5 Use the `nnmtrapconfig.ovpl` command to determine whether your expected trap is being blocked tool:

**`nnmtrapconfig.ovpl -dumpBlockList -u system -p mypassword`**

When examining the output, note the following:

- The allowed OIDs are from the SNMP traps configured in the NNMi incident configuration.
- The blocked OIDs are from the `nnmtrapd.conf` file.
- The caches are dynamic information.
- If a trap comes in and does not pass the filters, its OID is added to the blocked cache.

An abbreviated version of the typical output is shown here.

An easy way to check if a trap is allowed is to look in the first section `List of allowed trap OIDs` as highlighted in the following example. In our particular case, the OID .1.3.6.1.4.1.9.9.41.2.0.1 is not listed.

```
Filter Configuration:
List of allowed trap OIDs:
.1.3.6.1.4.1.9.9.13.3.0.6
.1.3.6.1.4.1.9.10.106.2.1
.1.3.6.1.4.1.2272.1.21.41

Blocking Caches:
Cache of blocked OIDs:
.1.3.6.1.6.3.1.1.5.5.1.3.6.1.4.1.9.1.400
.1.3.6.1.2.1.47.2.0.1
.1.3.6.1.4.1.9.0.1

Cache of trap OID's blocked from specific source addresses:
```

In our example, we determined the following:

- The SNMP trap is not blocked by the NNMi Trap Service.
- The SNMP trap is not in the allowed list.

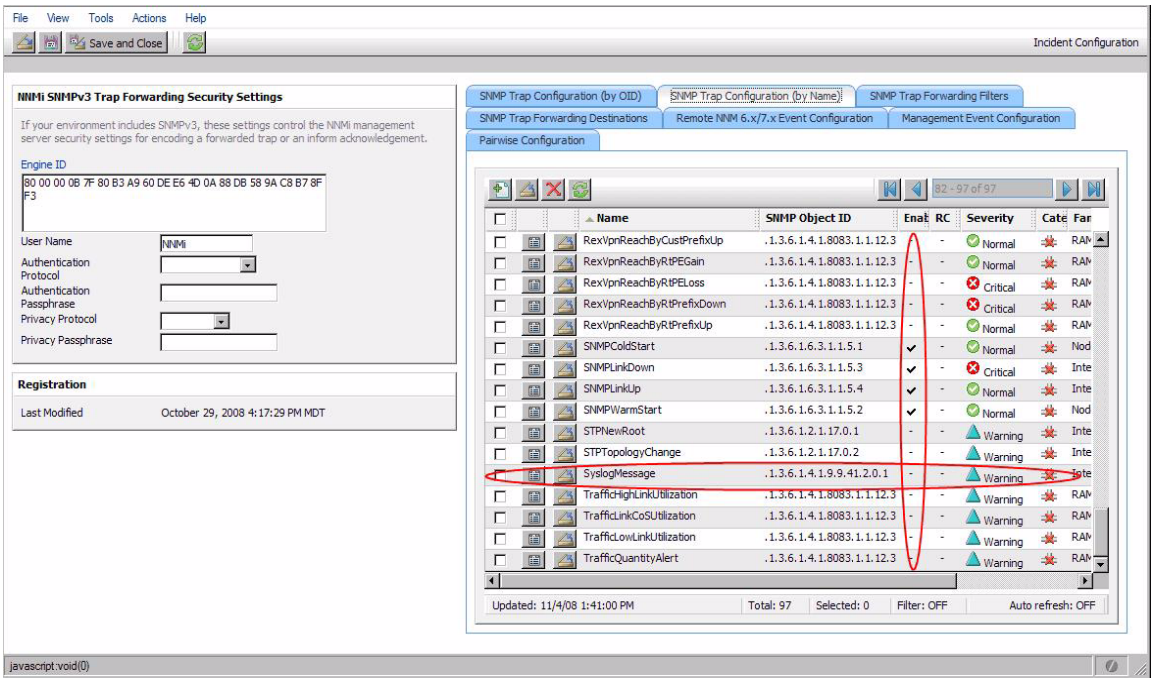
This means the SNMP trap of interest must be influenced by the NNMi incident configuration.

6 Navigate to the **Incident Configuration** form and look for the SNMP trap of interest:

a From the workspace navigation panel, select the **Configuration** workspace.

b Select **Incident Configuration**.

c Select the **SNMP Trap Configuration (by OID)** or **SNMP Trap Configuration (by Name)** tab.




Note the following:

- If the trap is not loaded, it is filtered by the NNMi Trap Service. You can load the trap using the `nnmincidentcfg.ovpl -loadTraps` command.
- If the trap is loaded, check whether the SNMP trap is enabled. If it is not enabled, it is filtered by the NNMi database.

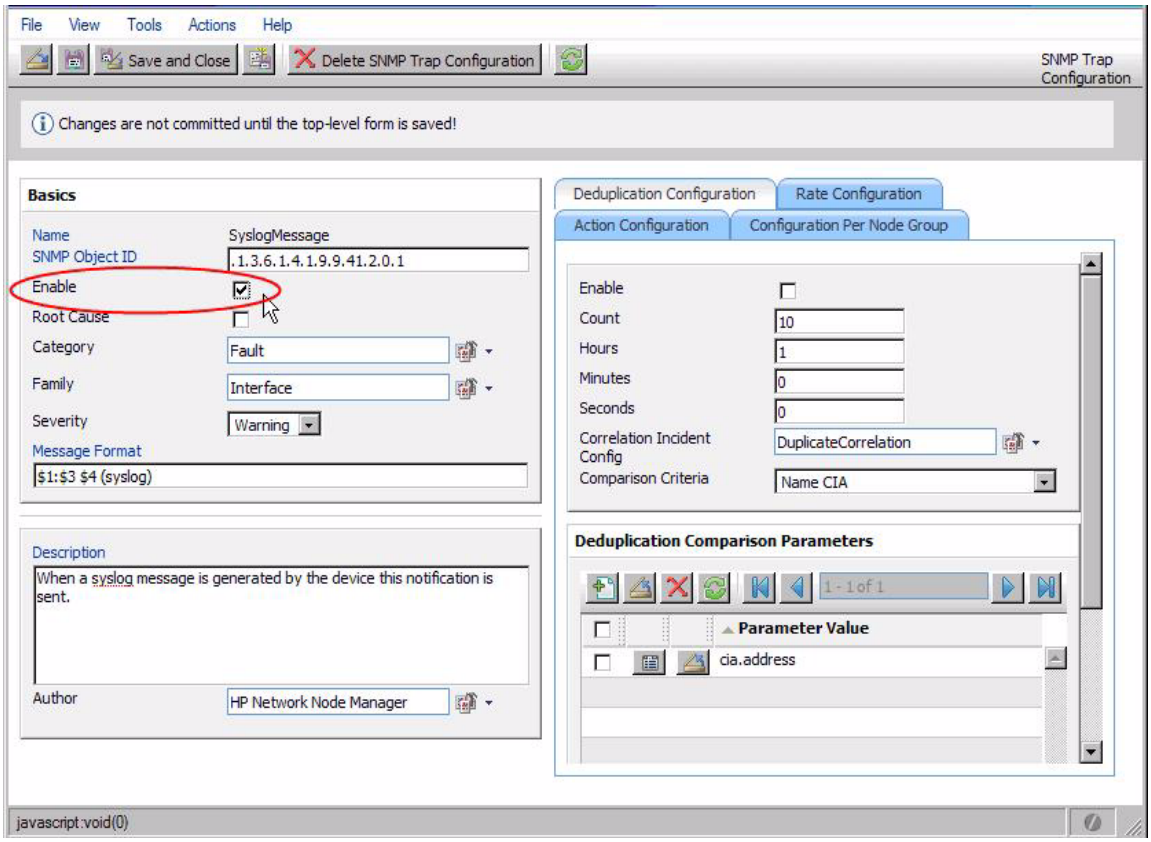
In our case, we see that the trap is loaded.

7 Check whether the SNMP trap is enabled:

a Select the `ciscoSyslogMIBNotification trap` configuration.

b Open the **SNMP Trap Configuration** form for the `ciscoSyslogMIBNotification` trap configuration and verify that the **Enable** box is selected.

As shown in the following example, the **Enable** attribute indicates that the SNMP trap configuration is enabled.




c Save and close the **SNMP Trap Configuration** form.

d Save and close the **Incident Configuration** form.

8 To confirm the change, use the `nnmtrapconfig.ovpl` tool to view the list of allowed SNMP trap OIDs:

```
nnmtrapconfig.ovpl -dumpBlockList -u system -p mypassword
```

You should see the OID for the SNMP trap that you just enabled.

```
List of allowed trap OIDs:
.1.3.6.1.4.1.9.9.13.3.0.6
.1.3.6.1.4.1.9.10.106.2.1
.1.3.6.1.4.1.2272.1.21.41
.1.3.6.1.4.1.9.9.41.2.0.1
```

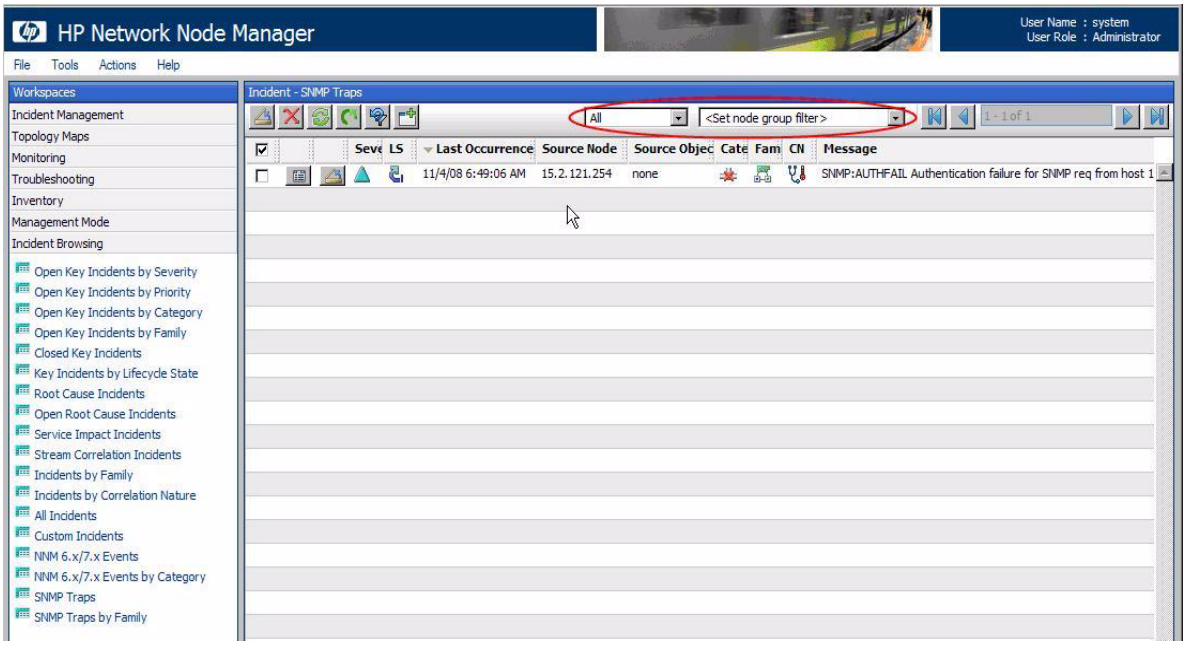
9 To make sure that the traps are arriving, navigate to the **SNMP Traps** view:

a From the workspace navigation panel, select the **Incident Browsing** workspace.

b Select **SNMP Traps**.

You should begin seeing the traps arrive.

As shown in the following example, the top of the **SNMP Traps** view includes a "time" filter and a Node Group filter. Make sure that these are set appropriately for your testing. Sometimes these filters are set to show only the last hour or a certain group of nodes that does not include the node of interest.




# Task 2: Manage the Number of Incoming SNMP Traps

This example steps through troubleshooting the following scenario:

You are getting too many traps into the NNMi console and want to reduce the number of incoming traps.

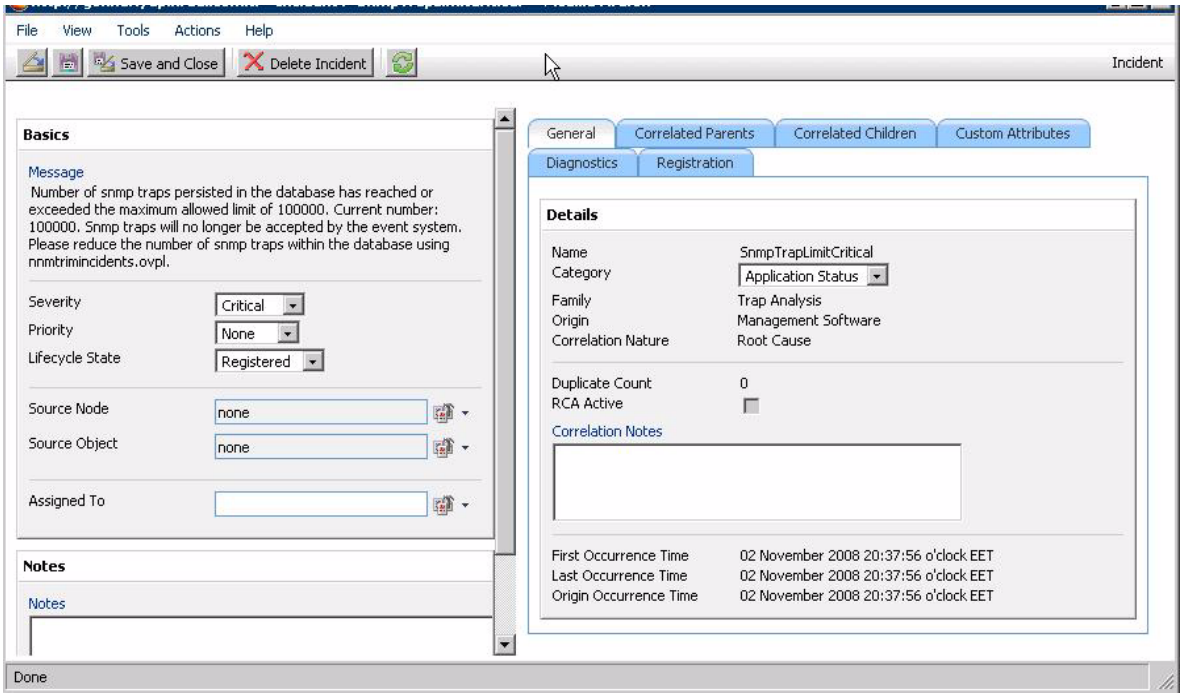
Analysis done against the SNMP Trap Service storage (Trap Store) can be different than what is shown in the NNMi console, which reflects the NNMi database contents. This scenario uses both the SNMP Trap Service storage (Trap Store) and the NNMi database. Therefore, pay close attention to whether the step described is using data from the NNMi database or from the SNMP Trap Service storage (Trap Store).

When working with SNMP trap limits, note the following:

- NNMi allows 100,000 traps to be stored in the NNMi database. The 100,000 trap limit applies only to the NNMi database. After this limit is reached, no additional traps are stored in the NNMi database.
- The NNMi Trap Service storage (Trap Store) continues to receive traps after the 100,000 SNMP Trap limit is reached. When the NNMi Trap Service storage has “filled up”, it uses the FIFO (First In, First Out) model, deleting the oldest traps and saving the newest ones.

When using results from the NNMi database, note the following:

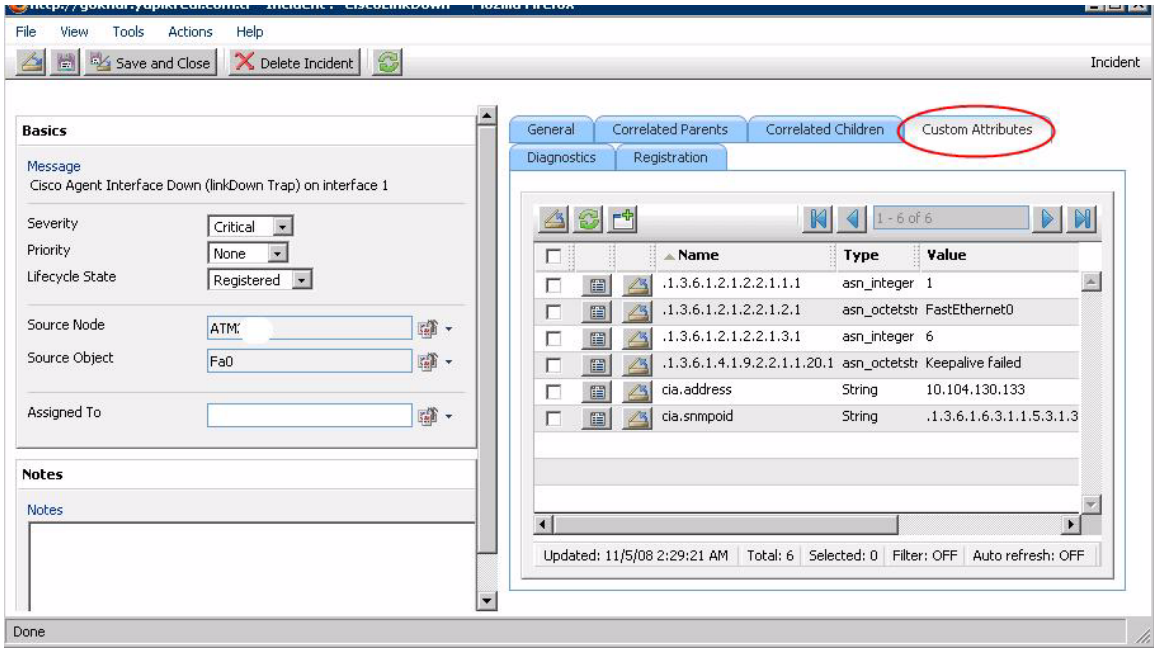
- Although the NNMi console provides a visual inspection of the traps stored in the database, use `nnmtrapdump.ovpl` and `nnmtrapconfig.ovpl` to analyze the data as described in this section of the document.
- Use caution when viewing SNMP trap information in the NNMi console. The Source Node displayed in the NNMi console is not necessarily the SNMP trap source address. Instead, the trap could be coming from any address on the Source Node.
- NNMi generates an incident when you are near or at the limit of total SNMP traps allowed in the NNMi database as shown in the following example:



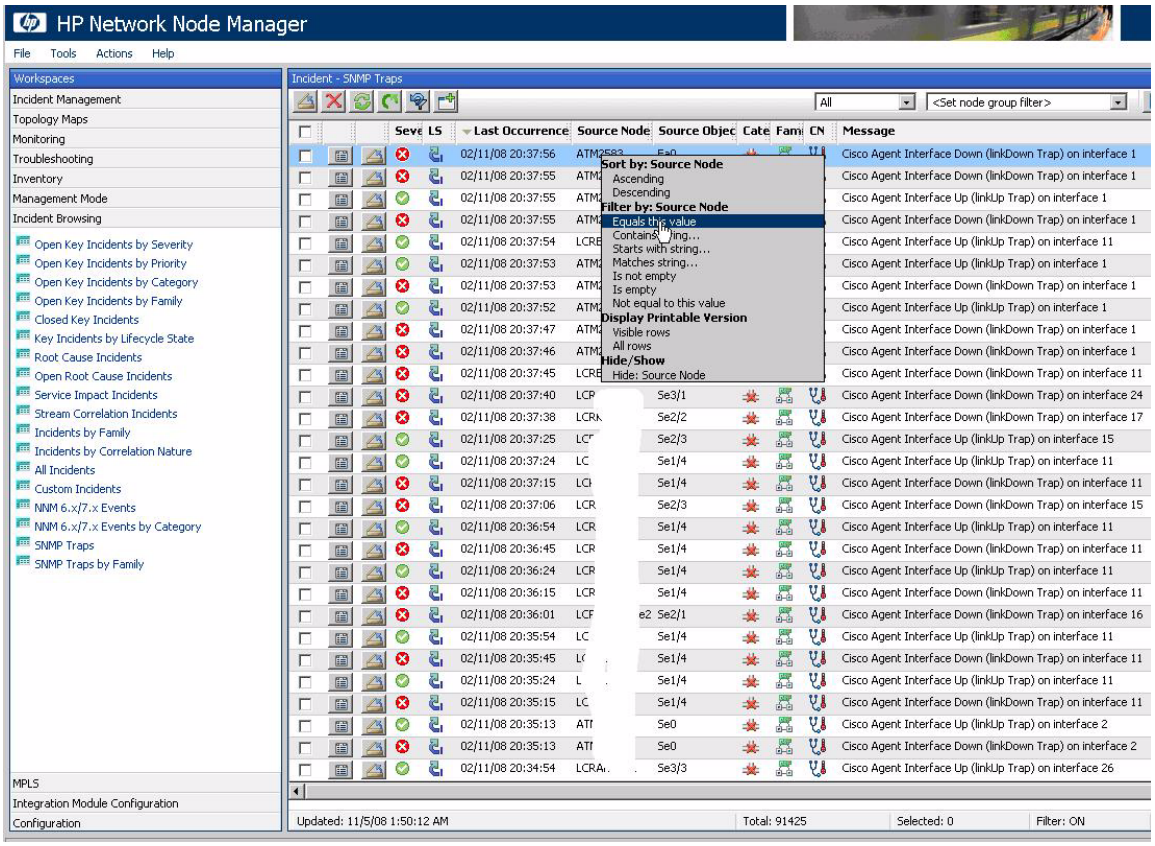

See **Task 3: Periodically Trim the NNMi Database of SNMP Traps** for information about how to remove traps from the database.

To manage the number of SNMP traps, start by determining which nodes are generating the most traps.

1. To determine which nodes are generating the most traps, begin by viewing information on the **Custom Attributes** tab for the SNMP Trap configuration of interest. The Custom Attributes tab is an excellent source of information about a particular trap in the NNMi console. This tab shows all of the information about a selected trap, including the Source Address as shown in the following example:




When looking at the SNMP Traps view, it can be difficult to get a sense of which nodes are generating the most SNMP traps. You can try the filters provided to see which Source Node is generating the most SNMP traps.



You can also use the `nnmtrimincidents.ovpl` command to export the trap information from the NNMi console to a comma separated file for easier analysis. This file can then be imported into a tool such as Microsoft Excel.

The following example generates a file with all traps greater than one day old that are in the NNMi database (not the NNMi Trap Service storage (Trap Store)):

```
nnmtrimincidents.ovpl -age 1 -incr days -origin SnmpTrap -
archiveOnly -u system -p mypassword
```

The `-archiveOnly` option does NOT trim incidents from the database.

When importing the file into Excel, note the following:

- Before importing the file, delete the first few lines of the output file that describe the file contents. (These lines begin with # .)
- Specify that the file is comma separated.

An example output file that can be used for analysis is shown below..

| | A | B | C | D | |
|---|---|---|---|---|---|
| 238 | Wed Feb 25 04:49:15 MST 2009 | 192.25.203.67 | none | Cisco incorrect community name (SNMP authenticationFail | R |
| 239 | Tue Feb 24 23:34:14 MST 2009 | 192.25.203.67 | none | Cisco incorrect community name (SNMP authenticationFail | R |
| 240 | Tue Feb 24 23:34:16 MST 2009 | 192.25.203.67 | none | Cisco incorrect community name (SNMP authenticationFail | R |
| 241 | Tue Feb 24 23:54:17 MST 2009 | 192.25.203.67 | none | Cisco Agent Interface Up (linkUp Trap) on interface 4 | R |
| 242 | Tue Feb 24 23:54:19 MST 2009 | 192.25.203.67 | none | Cisco Agent Interface Up (linkUp Trap) on interface 4 | R |
| 243 | Tue Feb 24 23:54:25 MST 2009 | 192.25.203.67 | none | Cisco Agent Interface Up (linkUp Trap) on interface 4 | R |
| 244 | Tue Feb 24 23:54:45 MST 2009 | 192.25.203.67 | none | Cisco Agent Interface Down (linkDown Trap) on interface 4 | R |
| 245 | Wed Feb 25 03:28:26 MST 2009 | 192.25.203.67 | none | Cisco incorrect community name (SNMP authenticationFail | R |

Alternatively, you can work with the archive file from the command line. For example, to get a list of the trap OID's:

```
# cut -f 20 -d ',' incidentArchive.txt | sort -rn | uniq -c
```

In the example output below, the left hand column represents the count.

```
254  .1.3.6.1.6.3.1.1.5.5.1.3.6.1.4.1.9
  3  .1.3.6.1.6.3.1.1.5.4.1.3.6.1.4.1.9
  8  .1.3.6.1.6.3.1.1.5.3.1.3.6.1.4.1.9
  4  .1.3.6.1.6.3.1.1.5.1.1.3.6.1.4.1.9
```

From this example, you can see that you might want to disable the Cisco Authentication Failure trap (.1.3.6.1.6.3.1.1.5.5.1.3.6.1.4.1.9).

To get a list of the source addresses and occurrence count:

```
# cut -f 23 -d ',' incidentArchive.txt | sort -rn | uniq -c
```

From the example output below, you can see that you might want to block traps from the node 10.25.203.67.

```
269  10.25.203.67
 10  10.18.10.2
```

The next two sections "Blocking Traps Using the Node IP Address" and "Blocking Traps Using the Trap OID" describe how to block traps using the `nnmtrapd.conf` file.

For the remainder of this example, instead of working with the SNMP trap information in the NNMi database, we analyze the SNMP traps in the NNMi Trap Service storage (Trap Store).

Advantages to analyzing the SNMP traps in the NNMi Trap Service storage include the following:

- The tools provide information that you cannot obtain using the NNMi database.
- You can determine which nodes and traps are the most verbose, even if they are being filtered at the NNMi Incident level.
- If you can discard the trap at the NNMi Trap Service level, NNMi performs more efficiently.

A disadvatage to using the NNMi Trap Service data is that you might get information about SNMP traps that are dropped before arriving to the NNMi console.

When using the NNMi Trap Service data, you can analyze your existing SNMP Trap information using the node's IP Address or the SNMP Traps OID.

## Blocking Traps Using the Node IP Address

1. Obtain a list of all the Source Addresses for SNMP traps in the NNMi Trap Service storage (Trap Store).
2. Send the contents of the NNMi Trap Service storage (Trap Store) to a file using the nnmtrapdump.ovpl tool:

   **nnmtrapdump.ovpl -u system -p mypassword > /tmp/trapdump.txt**
3. Find all occurrences of "Agent address" and delete extra words so that only the IP addresses are included. For example, on UNIX:

   **grep "Agent address" /tmp/trapdump.txt | cut -f 3 -d ' ' > /tmp/a**
4. Sort the IP addresses, obtain a unique count for each IP address, and sort the results in order of the most occurrences to least occurences. For example, on UNIX:

   **sort -rn /tmp/a | uniq -c | sort -rn > /tmp/trapcount.txt**

   The file `/tmp/trapcount.txt` should look similar to the following.

   The column on the left is the unique count. The column on the right is the source address.

```
60109 10.168.201.16
 5064 10.98.112.2
 5061 10.98.110.2
 1089 10.130.252.90
   63 10.168.129.19
   52 10.104.252.7
   36 10.106.252.19
   36 10.103.252.18
```

```
16 10.108.130.37
16 10.102.252.14
12 10.105.252.8
12 10.102.130.12
```

From this output, you can see that most traps are coming from the node 10.168.201.16.

5 Block the traps from node 10.168.201.16 by editing the blocking filters in the file `nnmtrapd.conf`. The location of this file is:

UNIX:

`/var/opt/OV/shared/nnm/conf/nnmtrapd.conf`

Windows:

`data_dir\shared\nnm\conf\nnmtrapd.conf`

This file contains examples and documentation.

When working with your own data, before blocking traps from the node, you would first evaluate the node and its traps to determine whether the traps are worth keeping.

1 To block all traps using the node's address, add the following line to the end of the `nnmtrapd.conf` file.

**`<10.168.201.16,.*>`**

Notice the use of wildcards **`(.*)`** to catch all traps from this source address.

2 Use the `nnmtrapconfig.ovpl` command to indicate the NNMi Trap Service should re-read the filter file:

**`nnmtrapconfig.ovpl -readFilter -u system -p mypassword`**

3 Confirm that the change took place:

**`nnmtrapconfig.ovpl -dumpBlockList -u system -p mypassword`**

You should see the following entry to confirm the traps are blocked from this source address:

```
List of blocked source addresses:
10.168.201.16
```

## Blocking Traps Using the Trap OID

You could do a similar process based on trap OID rather than source address.

1 Run the following commands to use OID numbers to determine the highest number of SNMP traps per OID being generated:

**`nnmtrapdump.ovpl -u system -p mypassword > /tmp/trapdump.txt`**

```
grep Trap /tmp/trapdump.txt | cut -f 2 -d ' ' > /tmp/b
sort -rn /tmp/b  | uniq -c | sort -rn > trapOID.txt
```

To continue this example, suppose that we determine that the trap OID .1.3.6.1.2.1.10.21.2.0.1 is occurring often, is filling the NNMi console with traps, and we do not want to receive this trap.

You can block the trap in one of two ways:

- Use the NNMi console.

  Assuming you have this trap defined in NNMi, navigate to the incident configuration view, find the definition of this trap and uncheck the enable box as described in **Task 1: Determine why SNMP Traps are not Appearing in NNMi Console**.

- Use the nnmtrapd.conf file

  Blocking the trap with this method causes the trap to still be received into the NNMi Trap Service storate (Trap Store). An advantage to this method is that you could consider only blocking this trap when it is received from a specific IP Address. This is not possible in the NNMi console.

  For our example, suppose you wish to block this trap when it is received from any address in the subnet 10.6.3.*.

2 Using the `nnmtrapd.conf` file to block any trap with an OID of .1.3.6.1.2.1.10.21.2.0.1 from any source node in the subnet 10.6.3.*, edit the `nnmtrapd.conf` file to include the line below:

```
<10.6.3.*, .1.3.6.1.2.1.10.21.2.0.1>
```

You can also mix and match IP address ranges and wildcards as well as OID ranges and wildcards.

## Additional Tips for Managing Your SNMP Trap Numbers

Troubleshooting Tip: Filter Placement

A common error when working with the `nnmtrapd.conf` file is to have two lines that match the same IP address. The parser works from the top to the bottom. When it finds an IP address match, the parser does not continue further.

For example, only the first line is effective if the `nnmtrapd.conf` file contains the following two lines:

```
<10.6.3.*, .1.3.6.1.2.1.14.16.0.6>
<10.6.3.*, .1.3.6.1.2.1.10.21.2.0.1>
```

Because the node address matches this first line, the second line is never matched.

To enable `nnmtrapd.conf` to match each specified filter, use syntax similar to the following:

```
<10.6.3.*, .1.3.6.1.2.1.14.16.0.6,.1.3.6.1.2.1.10.21.2.0.1>
```

As a general rule, use the NNMi console to block traps unless you only want to block them from specific IP addresses.

Best practice

To ensure all of the specific filters are enabled, include all the more specific filters above a general "catch all" filter at the bottom of the `nnmtrapd.conf` file. For example, if the `trapd.conf` file included an OID filter in our previous example, and it occurred below the IP address filters in the `trapd.conf` file, the OID filter might never be matched.

Troubleshooting Tip: Blocking Traps Using the NNMi Console

If you have the MIB loaded in the NNMi database for the traps that you wish to block, you could load the trap definition and then block the trap by simply disabling the trap in the NNMi console. Using the NNMi console does not reduce performance and you might find it easier to manage.

The benefit of using the `nnmtrapd.conf` file is that you get finer granularity for controlling traps because you can use wildcarding and specify nodes and SNMP trap OIDs together.

SNMP traps appear in the NNMi Trap Service storage even if they are blocked. Therefore, ignore statistics on traps that you know you are blocking.

# Task 3: Periodically Trim the NNMi Database of SNMP Traps

This section explains how to use the `nmtrimincidents.ovpl` command to trim the database of traps that are no longer needed. This practice helps you to avoid reaching the trap limit in the NNMi database.

The `nmtrimincidents.ovpl` command applies to the NNMi database and not the NNMi Trap Service storage (Trap Store).

The `nnmtrimincidents.ovpl` command has a rich set of options to help you trim the database.

Best practice

Run the `nnmtrimincidents.ovpl` command regularly in a cron job (UNIX) or a scheduled task (Windows).

The following set of nnmtrimincidents.ovpl deletes any traps that are older than one week.

```
nnmtrimincidents.ovpl -u system -p mypassword -age 1 -incr weeks
-origin SnmpTrap -trimOnly -quiet
```

You can adjust the example to fit your trap rate.

When using `nnmtrimincidents.ovpl`, note the following:

- The example options do not keep an archive of the trimmed traps. You can easily do this using the `-archiveOnly` option. See the `nnmtrimincidents.ovpl` reference page `(Help --> Documentation Library --> Reference Pages)` for more information .
- The `-quiet` mode does not prompt for user confirmation. (This option is ideal for a cron job.)


invent


Java™
COMPATIBLE