

How to Use SNMP Traps with NXTI in Nagios XI 5

Purpose

This document is intended for use by Nagios Administrators who would like to know how to use the Nagios XI SNMP Trap Interface (NXTI) to monitor and manage incoming SNMP Traps.

NOTICE: This article applies to Nagios XI version 5.x. For Nagios XI version 2024, please see this [article](#).

Overview

NXTI was introduced with Nagios XI 5.5 and is enabled with the Enterprise edition license of Nagios XI. If you do not have the Enterprise edition license you can still configure Nagios XI to accept SNMP traps; however, this is not covered here, please refer to the [Integrating SNMP Traps With Nagios XI](#) documentation.

NXTI serves as a web front-end to the `snmptrapd/snmpd` workflow configuration. Previously this was configured by manually editing configuration files (still the required method if you don't have the Enterprise edition license). NXTI additionally provides reporting of the traps received which are stored in a database. This functionality is exclusive to NXTI.

What Is an SNMP Trap?

The following is taken from the [Net-SNMP](#) website:

“A trap is a SNMP message sent from one application to another (which is typically on a remote host). Their purpose is merely to notify the other application that something has happened, has been noticed.”

An important point to stress with SNMP traps is that they are asynchronous events that can occur at any time. In Nagios XI this is what is called a **Passive** check/service. This means that they are not actively checked by Nagios XI on a regular schedule. Nagios XI is waiting for a SNMP Trap to be received from the remote device. A comparison between an active check and a passive check helps explain the difference between **Active** and **Passive** checks:

Scenario: A UPS device loses input power and is running on batteries.

- With an **Active** check, if Nagios XI was checking the device on a 5-minute interval, then it might be up to 5 minutes before Nagios XI is aware that the device is running on batteries.
- With a **Passive** check, the device immediately sends an SNMP Trap to Nagios XI when it is running on batteries

More detailed information on passive services can be found in our [How to Configure Passive Services with Nagios XI](#) documentation.

How to Use SNMP Traps with NXTI in Nagios XI 5

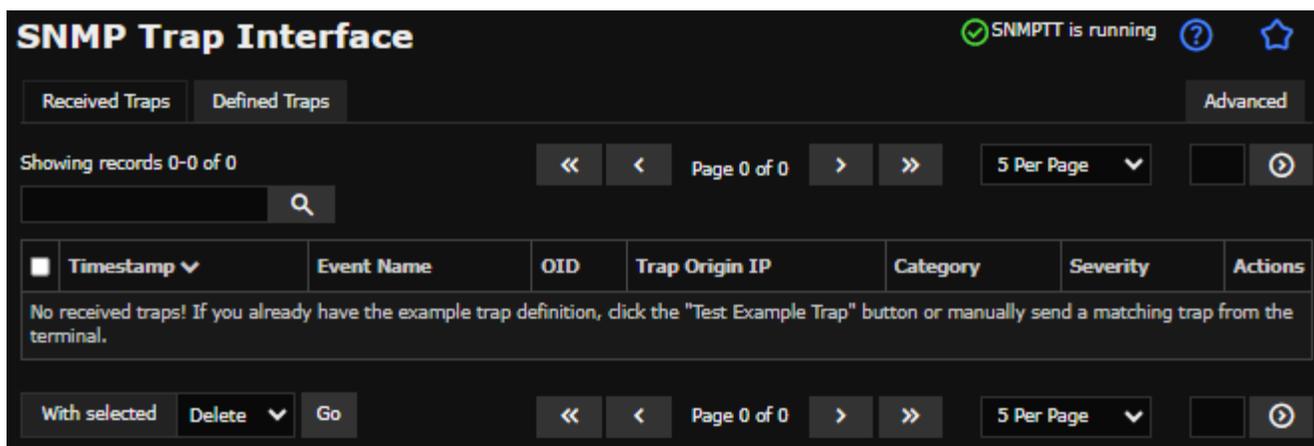
SNMP v2 vs SNMP v3

SNMP traps can be received using v2 or v3 of the protocol. By default, the Nagios XI server will accept inbound SNMP v2 traps from any device. Security for accepting SNMP v2 traps is explained in our [Nagios XI - SNMP Trap Hardening](#) article.

Nagios XI needs to be configured before it can accept SNMP v3 traps, this is detailed in our [Nagios XI - SNMP Trap v3 Configuration](#) article.

NXTI Interface

To access NXTI navigate to **Admin > Monitoring Config > SNMP Trap Interface**.



The screenshot displays the 'SNMP Trap Interface' in Nagios XI. At the top, there's a status indicator 'SNMPTT is running' with a green checkmark. Below this, there are two tabs: 'Received Traps' (selected) and 'Defined Traps'. An 'Advanced' button is located in the top right corner. The main content area shows 'Showing records 0-0 of 0' and a search bar. Below the search bar is a table with the following columns: 'Timestamp' (with a dropdown arrow), 'Event Name', 'OID', 'Trap Origin IP', 'Category', 'Severity', and 'Actions'. The table is currently empty, with a message: 'No received traps! If you already have the example trap definition, click the "Test Example Trap" button or manually send a matching trap from the terminal.' At the bottom of the table, there are buttons for 'With selected', 'Delete' (with a dropdown arrow), and 'Go'. The interface also includes pagination controls (Page 0 of 0) and a '5 Per Page' dropdown.

NXTI provides the following capabilities:

- View, Add, Edit, Copy, Delete, and Disable trap definitions.
- View and Delete received trap logs.
- Search and sort both trap definitions and received trap logs.
- Monitor the *snmpd* process.
- Locally test *snmptrapd/snmpd* functionality.

NOTE: The SNMP Trap interface is only accessible to Nagios admins. Non-admin users will not have access.

How NXTI Works

NXTI utilizes the SNMPTT application that is provided with Nagios XI. This is how the operating system processes the received traps into useful data. If you have previously worked with the SNMPTT

How to Use SNMP Traps with NXTI in Nagios XI 5

configuration files, then you will be aware that it can become quite complex. NXTI provides a simple way to add, edit or remove trap definitions to the SNMPTT configuration.

The default SNMPTT configuration file on your Nagios XI server is `/etc/snmp/snmptt.conf` and is where the non-NXTI trap configurations reside. NXTI utilizes the separate configuration file `snmptt.conf.nxti`. This file should never be edited manually as those changes will be lost. Whenever you add, edit or remove a trap in NXTI, `snmptt.conf.nxti` is updated automatically and the `snmptt` service is restarted.

The trap definitions created by NXTI adhere to the [SNMPTT configuration file format](#), hence this documentation will explain how the NXTI fields relate to the SNMPTT trap definitions in the configuration file.

Every trap that is defined in a SNMPTT configuration file begins with an *EVENT* line. This is how the incoming trap is matched with a trap definition. Below is an example:

```
EVENT NXTI_Event_1 NET-SNMP-EXAMPLES-  
MIB::netSnmExampleHeartbeatNotification "NXTI Test Event" Normal
```

If an incoming trap is matched against the OID in the *EVENT* line, then the *EXEC* line(s) defined are executed (along with the other optional features of SNMPTT). The *EXEC* lines are how incoming trap data is actioned.

The most basic functionality that NXTI provides is to store a received trap in the database. These traps can be queried at any time after they have been received. While this basic configuration does not provide you with notifications for the received traps, it does allow you to receive a broad range of trap data that you can analyze without generating unnecessary notifications. This data is added to the database with the following *EXEC* command (just a sample of the line is shown):

```
EXEC php /usr/local/nagiosxi/scripts/nxti.php --event_name="$N" ....
```

To receive notifications for received traps, you can use the **Passive Service Setup** component of the trap definition. This adds an *EXEC* command like the sample of the line is shown below:

```
EXEC /usr/local/bin/snmptraphandling.py "$aR" "SNMP Traps" ....
```

Furthermore, you can define additional *EXEC* commands. This allows you to take other required actions for the received trap. Here is an example where a line is appended to a text file:

```
EXEC echo 'Success!' >> /usr/local/nagiosxi/var/NXTI_Write_Test
```

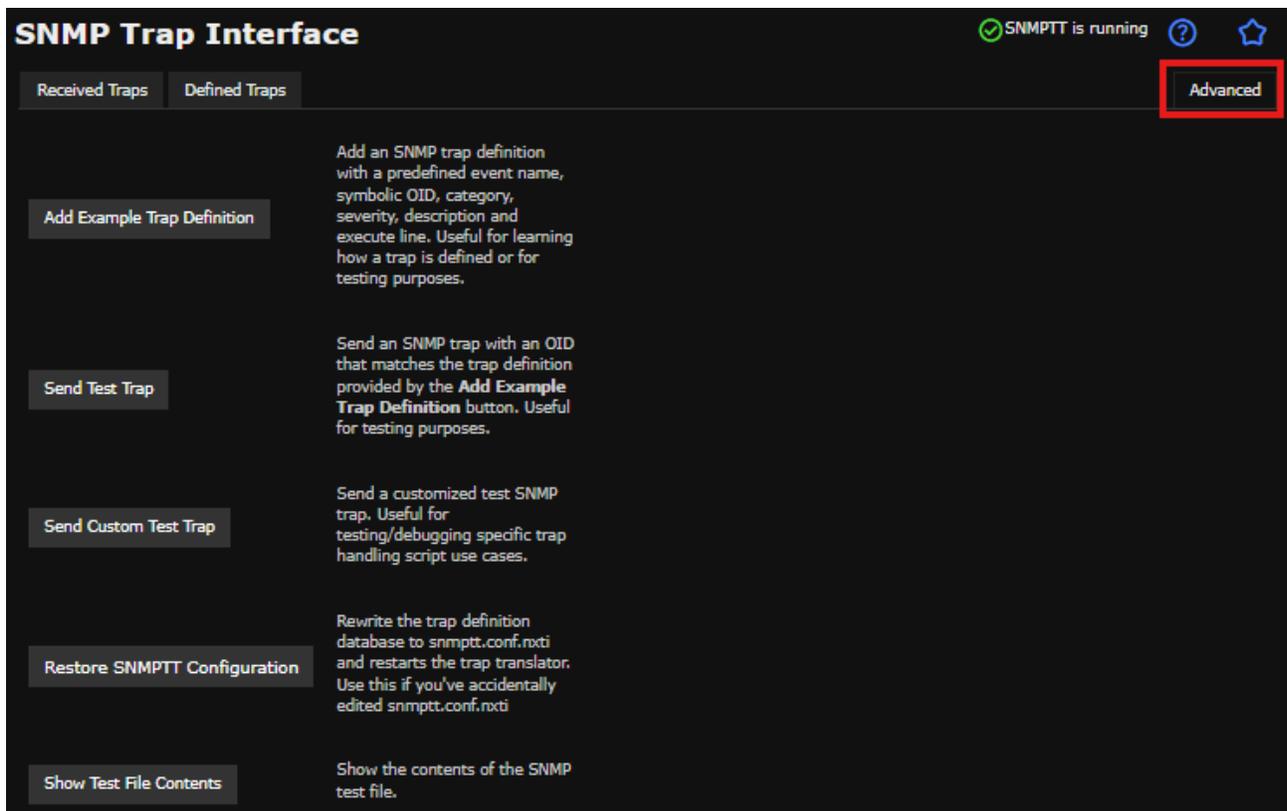
How to Use SNMP Traps with NXTI in Nagios XI 5

Finally, the advanced capabilities of SNMPTT can be defined such as PREEEXEC, NODES, MATCH, REGEX. These are outside the scope of this documentation, however, our [Nagios XI – SNMP Trap Tutorial](#) guide does explain how **MATCH** can be utilized in further detail.

Adding Trap Definitions

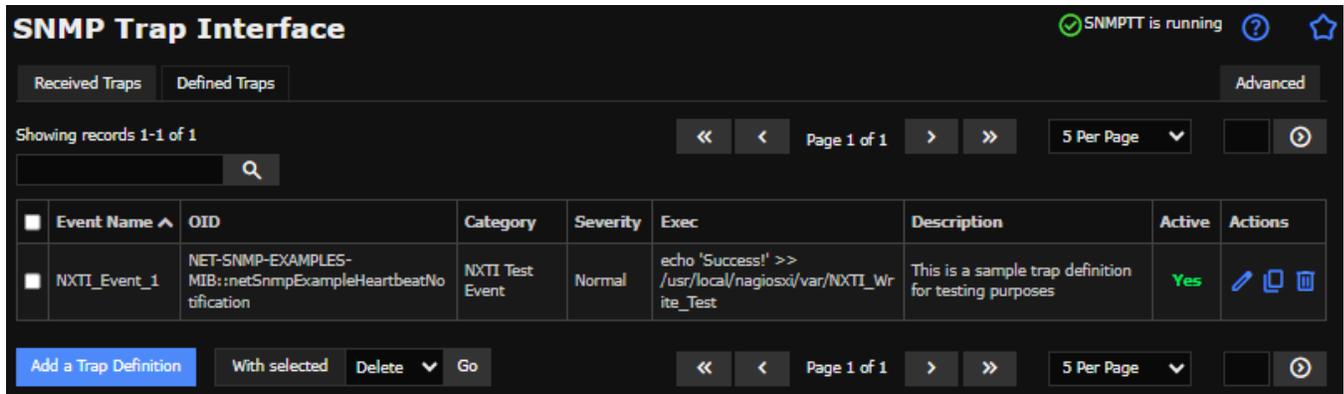
The **Defined Traps** tab allows you to create a trap definition and has many options available. For a beginner, these options can be overwhelming, and an example really helps learn how it works.

1. Click the **Advanced** tab which provides an **Add Example Trap Definition** button.



2. Once you click the **Add Example Trap Definition** button, a message will appear at the top of the screen telling you it was added.
3. Click the **Defined Traps** tab again to see the newly added trap definition.

How to Use SNMP Traps with NXTI in Nagios XI 5



The screenshot shows the 'SNMP Trap Interface' in Nagios XI. It has tabs for 'Received Traps' and 'Defined Traps'. The 'Defined Traps' tab is active, showing a table with one record. The table has columns for Event Name, OID, Category, Severity, Exec, Description, Active, and Actions. The record for 'NXTI_Event_1' has an OID of 'NET-SNMP-EXAMPLES-MIB::netSnmExampleHeartbeatNotification', a category of 'NXTI Test Event', a severity of 'Normal', and an exec command that runs a PHP script and echoes 'Success!' to a file. The 'Active' status is 'Yes'.

Event Name	OID	Category	Severity	Exec	Description	Active	Actions
NXTI_Event_1	NET-SNMP-EXAMPLES-MIB::netSnmExampleHeartbeatNotification	NXTI Test Event	Normal	echo 'Success!' >> /usr/local/nagiosxi/var/NXTI_Write_Test	This is a sample trap definition for testing purposes	Yes	  

- The trap that was added demonstrates how you can append some text to a file on the Nagios XI server when a heartbeat trap is received.
- Click the **edit** icon in the **Actions** column to edit the trap. This opens the **Edit Trap Definition** page, which is almost identical to the **Add a Trap Definition** tab.

Here is that example trap definition as it exists in the `snmptt.conf.nxti` file.

```
EVENT NXTI_Event_1 NET-SNMP-EXAMPLES-
MIB::netSnmExampleHeartbeatNotification "NXTI Test Event" Normal
FORMAT Received trap "$N" with variables "$+*"
EXEC php /usr/local/nagiosxi/scripts/nxti.php --event_name="$N"
--event_oid="$i" --numeric_oid="$o" --symbolic_oid="$O" --community="$C"
--trap_hostname="$R" --trap_ip="$aR" --agent_hostname="$A"--agent_ip="$aA"
--category="$c" --severity="$s" --uptime="$T" --datetime="$x $X" --
bindings="$ $ +*"
EXEC echo 'Success!' >> /usr/local/nagiosxi/var/NXTI_Write_Test
DESC
```

This is a sample trap definition for testing purposes

```
EDESC
```

There are four components to adding a trap definition:

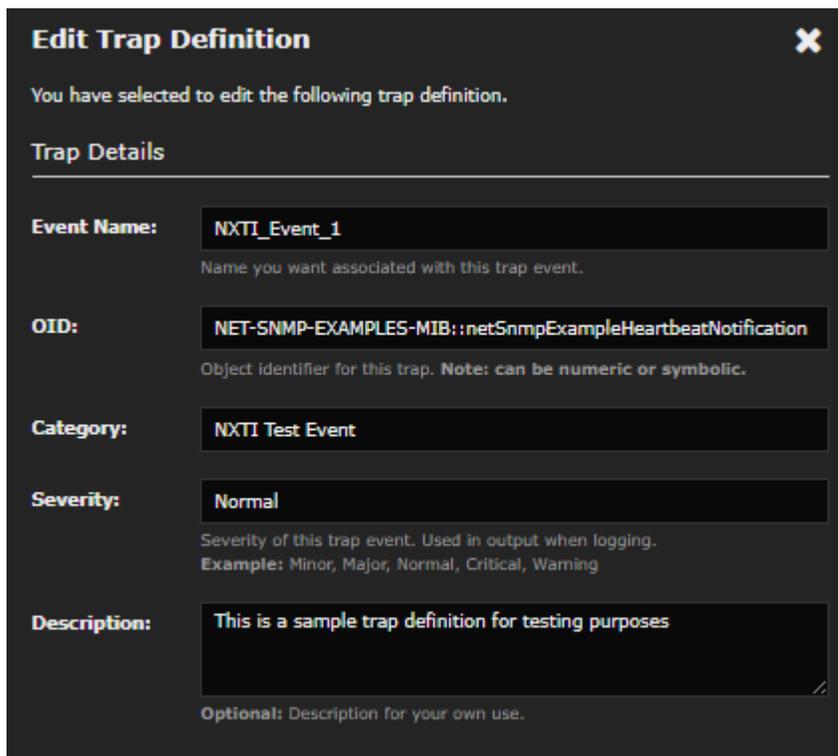
- [Trap Details](#): This is how a trap is identified and classified.
- [Passive Service Setup](#): Configure the trap so that Nagios XI can receive the trap data in a service.
- [Exec](#): Defines commands to be executed when the trap is received.

How to Use SNMP Traps with NXTI in Nagios XI 5

- [Advanced](#): For the advanced capabilities of SNMPTT such as PREEEXEC, NODES, MATCH, REGEX.

Trap Details

The fields **Event Name**, **OID**, **Category** and **Severity** are specifically for the EVENT line in a trap definition and are required.



Edit Trap Definition ✕

You have selected to edit the following trap definition.

Trap Details

Event Name:
Name you want associated with this trap event.

OID:
Object identifier for this trap. Note: can be numeric or symbolic.

Category:

Severity:
Severity of this trap event. Used in output when logging.
Example: Minor, Major, Normal, Critical, Warning

Description:
Optional: Description for your own use.

These directives are mandatory:

- Event Name
 - This must be a unique name that cannot contain spaces.
- OID
 - This is how an incoming trap is matched against this trap definition.
 - Can be either a full numeric OID or a symbolic OID.
 - Numeric OID is the raw (hard to read) format, for example:
 - .1.3.6.1.4.1.8072.2.3.0.1

How to Use SNMP Traps with NXTI in Nagios XI 5

- Symbolic OID is an easy-to-read version of the numeric OID (case-sensitive).
- You can define the fully qualified MIB name, for example:
 - `iso.org.dod.internet.private.enterprises.netSnmp.netSnmpExamples.netSnmpExampleNotifications.netSnmpExampleNotificationPrefix.netSnmpExampleHeartbeatNotification`
- You can also use a shorter variant, for example:
 - `NET-SNMP-EXAMPLES-MIB::netSnmpExampleHeartbeatNotification`
- Category
 - Allows you to categorize the incoming trap as per your requirements
 - Options like **IGNORE** and **LOGONLY** should be avoided ([see documentation](#)).
- Severity
 - Typically has a value from one of: "Minor", "Major", "Normal", "Critical", "Warning."
 - It cannot contain spaces.
 - This field value correlates to the Passive Service Setup severity option of **Pass Severity Level**.
- The **Description** field corresponds to the text between the *SDESC* and *EDESC* lines.
 - This is used to describe the conditions and handling of the event to technicians.
 - This can contain any text.

The screenshot shows the 'Passive Service Setup' configuration page in Nagios XI. It includes a header 'Passive Service Setup' and a note: 'These inputs will also work with the EXEC macros table.' Below this is a checkbox labeled 'Enable Passive Service Setup' which is checked. The configuration fields are: 'Host Name' with the value '\$aR', 'Description' with the value 'SNMP Traps', 'Severity' with a dropdown menu set to 'Parse Severity Level (\$s)', and 'Output' with the value 'SNMP Trap Received at \$@ with variables \$+*'. Each field has a small icon to its right, likely for copying or clearing the text.

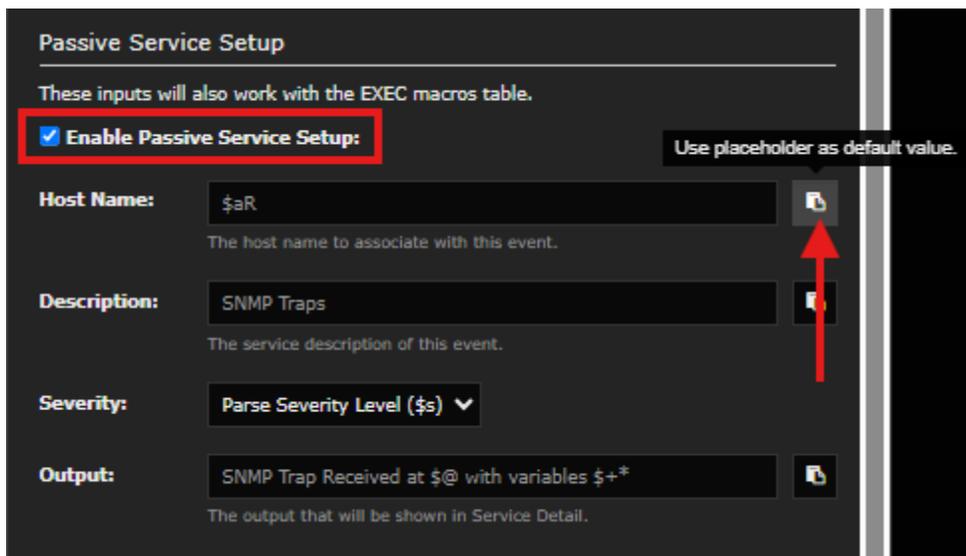
How to Use SNMP Traps with NXTI in Nagios XI 5

- This is optional.

Passive Service Setup

This section is how you can configure this trap definition to send a passive check result to a Nagios service. The point of doing this is so that your Nagios XI users can receive notifications when traps are received for this trap.

1. Check the box to enable this functionality. You will also be required to populate each field.
2. The values already provided in each field are sufficient; however, click the **placeholder** icon to the right of the field to use it.



Passive Service Setup

These inputs will also work with the EXEC macros table.

Enable Passive Service Setup:

Host Name:  Use placeholder as default value.
The host name to associate with this event.

Description: 
The service description of this event.

Severity: 

Output: 
The output that will be shown in Service Detail.

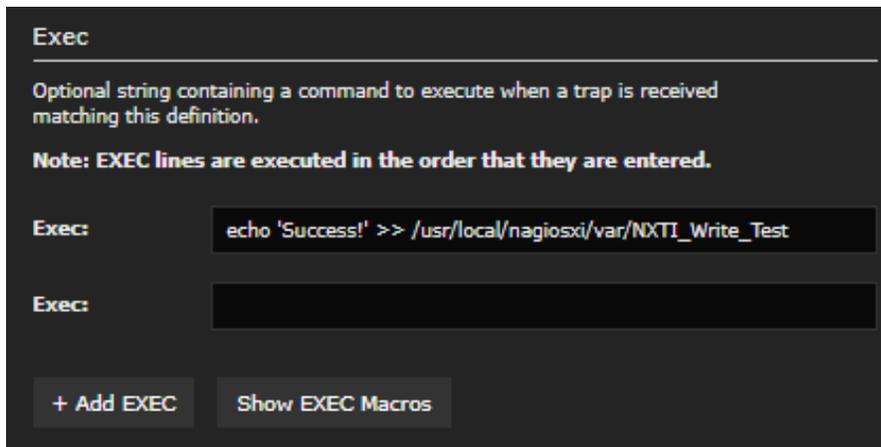
- a. The **Severity** selection designates what will be passed to Nagios as the state of the passive check result.
 - b. The **Pass Severity Level** option is commonly used as it does not hard code the state into the definition. This allows for advanced configurations where you have multiple trap definitions with identical OIDs but are using MATCH options to differentiate them. An example of this can be found in our [SNMP Trap Tutorial](#) article.
3. When you click **Enable Passive Service Setup**, an *EXEC* line is added to *snmptt.conf.nxti* for this definition.

How to Use SNMP Traps with NXTI in Nagios XI 5

Exec

Each Exec entry corresponds to an *EXEC* line. These are the commands that get executed by the *snmptt* process when each event occurs.

1. By clicking the **Show EXEC Macros** button at the bottom of the Exec entry, you can see the list of macros that can be used (these are built into SNMPTT).
2. You can add as many *EXEC* lines as required, simply click the **+ Add EXEC** button to make another field appear. **Removing** an *EXEC* line is done by clearing the contents of the field.



The screenshot shows the 'Exec' configuration section in Nagios XI. It includes a title 'Exec', a description 'Optional string containing a command to execute when a trap is received matching this definition.', and a note: 'Note: EXEC lines are executed in the order that they are entered.' Below this, there are two 'Exec:' labels followed by input fields. The first field contains the command 'echo 'Success!' >> /usr/local/nagiosxi/var/NXTI_Write_Test'. The second field is empty. At the bottom, there are two buttons: '+ Add EXEC' and 'Show EXEC Macros'.

Advanced

The Advanced section allows you to use other features of SNMPTT such as PREEEXEC, NODES, MATCH, REGEX.

These are commonly used to manipulate the trap data received before the *EXEC* lines are executed. A detailed example using a *MATCH* can be found in our [SNMP Trap Tutorial](#) article.

How to Use SNMP Traps with NXTI in Nagios XI 5

1. Clicking the **Save** button will apply the changes to the `snmptt.conf.nxti` file and restart the `snmptt` service.
2. Clicking the **Cancel** button will discard any changes you have made.

The screenshot shows two parts of the Nagios XI interface. The top part is an 'Advanced' configuration panel with a text area for 'Additional Raw Data' and 'Save' and 'Cancel' buttons. The bottom part is the 'SNMP Trap Interface' showing a table of defined traps. The 'Defined Traps' tab is highlighted with a red box. The table has columns for Event Name, OID, Category, Severity, Exec, Description, Active, and Actions. A single trap definition is visible.

Event Name ^	OID	Category	Severity	Exec	Description	Active	Actions
NXTI_Event_1	NET-SNMP-EXAMPLES-MIB::netSnmpExampleHeartbeatNotification	NXTI Test Event	Normal	echo 'Success!' >> /usr/local/nagiosxi/var/NXTI_Write_Test	This is a sample trap definition for testing purposes	Yes	

Managing Trap Definitions

The **Defined Traps** tab is how you manage your existing trap definitions.

By default, only 5 traps are displayed per page. This can be changed by using the **Per Page** drop down list that appears on the right-hand side of the table (top and bottom).

The buttons on either side of the page number count allow you to navigate back and forward through the pages.

You can jump to a specific page number by typing the number in the far-right field and clicking the **Jump to Page** button.

A close-up of the pagination controls. It shows navigation arrows, 'Page 1 of 1', '5 Per Page' dropdown, a text input field containing the number '3', and a 'Jump to Page' button. The text input field and the 'Jump to Page' button are highlighted with a red box.

How to Use SNMP Traps with NXTI in Nagios XI 5

When you have many defined traps, you can use the search field to find what you are after, this searches the **Event Name**, **OID** and **Description** fields.



The screenshot shows the 'SNMP Trap Interface' with tabs for 'Received Traps' and 'Defined Traps'. A search bar contains the text 'NET'. Below the search bar is a table with the following data:

Event Name	OID	Category
NXTI_Event_1	NET-SNMP-EXAMPLES-MIB::netSnmExampleHeartbeatNotification	NXTI Test Event

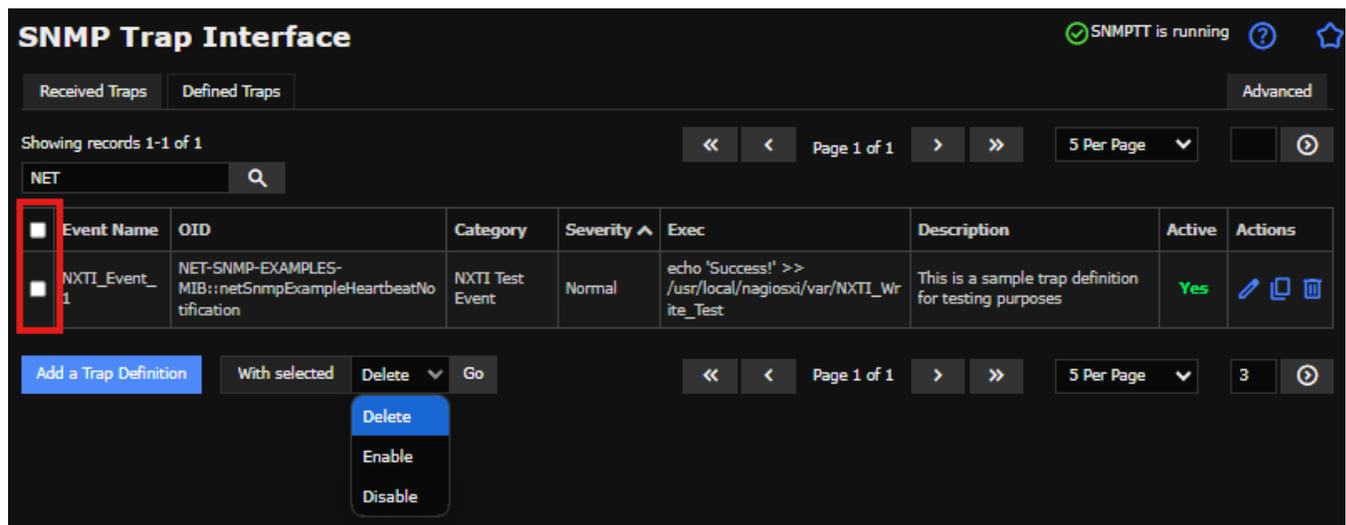


The screenshot shows the table headers for the trap definitions:

Event Name	OID	Category	Severity	Exec
------------	-----	----------	----------	------

The columns headings can be clicked on to sort the trap definitions as per your requirements.

1. In the **Actions** column you can click the **edit**, **copy** or **delete** a trap by clicking the associated icons.
2. The **Active** column allows you to disable a trap definition without deleting it.
 - a. When it is enabled, click the **Yes** word to disable it. This will turn to a **No**.
 - b. To enable the trap again, click the **No** word.
3. You can **Delete**, **Enable** or **Disable** multiple trap definitions at once by using the left column field to select multiple traps and using the **With selected** drop down menu.



The screenshot shows the 'SNMP Trap Interface' with a table of trap definitions. The table has columns for Event Name, OID, Category, Severity, Exec, Description, Active, and Actions. The 'Active' column shows 'Yes' for the selected trap. A dropdown menu is open under the 'With selected' button, showing options for 'Delete', 'Enable', and 'Disable'.

Event Name	OID	Category	Severity	Exec	Description	Active	Actions
NXTI_Event_1	NET-SNMP-EXAMPLES-MIB::netSnmExampleHeartbeatNotification	NXTI Test Event	Normal	echo 'Success!' >> /usr/local/nagiosxi/var/NXTI_Write_Test	This is a sample trap definition for testing purposes	Yes	  

How to Use SNMP Traps with NXTI in Nagios XI 5

4. Click **Go** to perform the requested action.

Managing Received Traps

The **Received Traps** tab is where you can report on all received traps that have been defined.

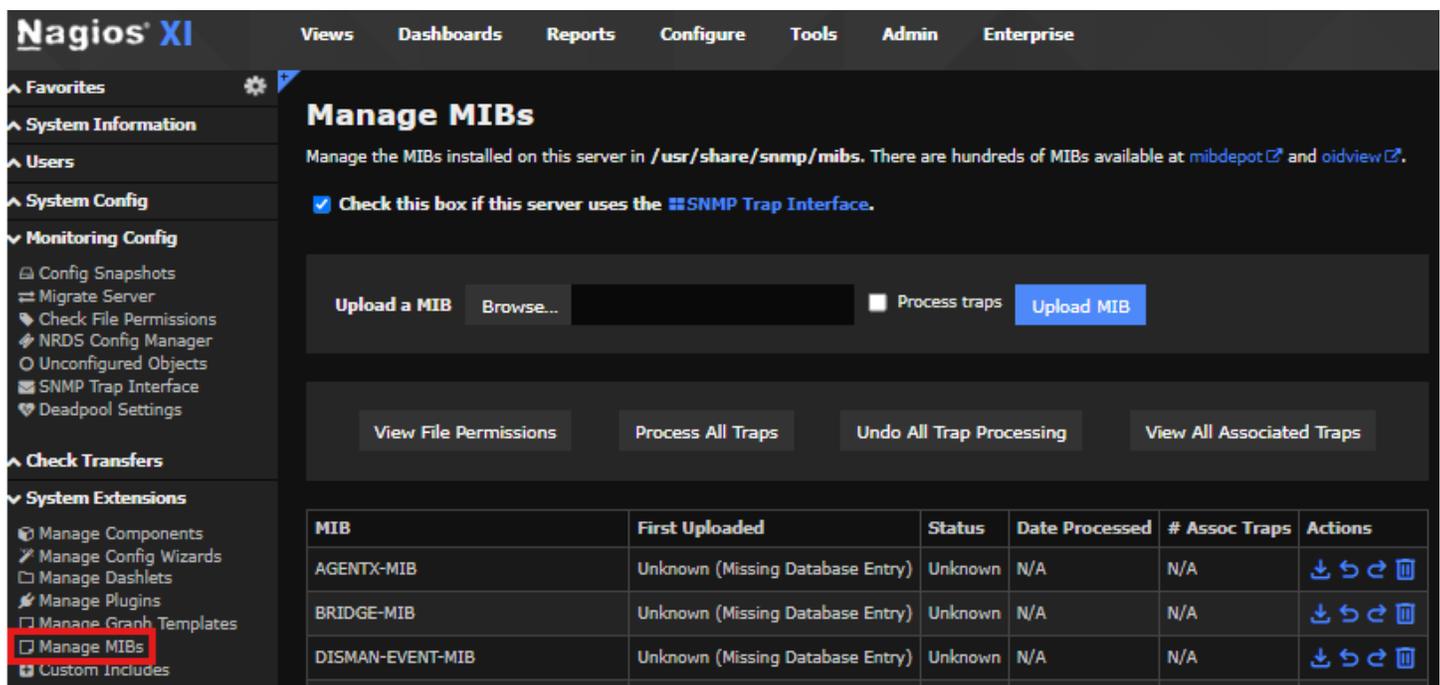
The controls available on this page are almost identical to the **Defined Traps** tab. You can change the number of records per page, search the records and delete (one or many).

Importing Trap Definitions

It's most likely that you will obtain a MIB file for your device that will include trap definitions. These can be imported into NXTI. There are several methods available for importing trap definitions into NXTI. We will explore each method in detail below.

Uploading A MIB File

You can upload a MIB file into Nagios XI via the **Admin > System Extensions > Manage MIBs** page.



The screenshot shows the Nagios XI interface for managing MIBs. The sidebar on the left includes sections for Favorites, System Information, Users, System Config, Monitoring Config, Check Transfers, and System Extensions. The 'Manage MIBs' page title is prominently displayed. Below the title, there is a checkbox labeled 'Check this box if this server uses the SNMP Trap Interface.' which is checked. The main content area features an 'Upload a MIB' section with a 'Browse...' button and an 'Upload MIB' button. Below this, there are buttons for 'View File Permissions', 'Process All Traps', 'Undo All Trap Processing', and 'View All Associated Traps'. A table lists existing MIBs with columns for MIB, First Uploaded, Status, Date Processed, # Assoc Traps, and Actions.

MIB	First Uploaded	Status	Date Processed	# Assoc Traps	Actions
AGENTX-MIB	Unknown (Missing Database Entry)	Unknown	N/A	N/A	↓ ↺ ↻ 🗑️
BRIDGE-MIB	Unknown (Missing Database Entry)	Unknown	N/A	N/A	↓ ↺ ↻ 🗑️
DISMAN-EVENT-MIB	Unknown (Missing Database Entry)	Unknown	N/A	N/A	↓ ↺ ↻ 🗑️

1. First, use the **Browse** button to locate a MIB file you have downloaded, select it in your files, and click **Upload MIB** button.
2. Once complete you can navigate to NXTI and locate the newly imported MIBs.

How to Use SNMP Traps with NXTI in Nagios XI 5

XI lets you choose between two systems for handling traps.

- Built-in SNMP trap interface.
- Legacy trap handling system in XI.

NOTE: If you wish to use the legacy system, you will need to deselect the **Check this box if this server uses the SNMP Trap Interface** checkbox.

Process Individual Traps

You can process an individual trap by checking the **Process traps** checkbox prior to uploading the MIB. If you forgot to select the box – don't worry, you can still process the trap by clicking on the **Process Traps** actions button (right blue arrow).

1. To see which traps have been processed on a MIB-by-MIB basis, click on the number, shown in the **# Assoc Traps** column.
2. To undo trap processing on a MIB-by-MIB basis, click on the **Undo Trap Processing** actions button.
3. You can download a MIB from your Nagios XI server to your workstation by clicking on the **Download** actions button (diskette icon).
4. You can also delete a MIB by clicking the **Delete** actions button (red X icon)

Process All Traps

You can process traps from all MIBs installed on your system by pressing the **Process All Traps** button.

This will process each MIB file in the `/usr/share/snmp/mibs` directory. If a trap exists it will import it into NXTI. Additionally, it will also double check the `/etc/snmp/snmpd.conf` file to see if the trap is already defined.

If it is defined, then it will comment it out of the `snmpd.conf` file so there is no duplicate definition.

1. You can see all the traps processed so far by clicking **View All Associated Traps**.
2. You can undo all the traps that have been processed by clicking **Undo All Trap Processing**.

Watch a video tutorial on how to upload and manage MIBs for SNMP in Nagios XI here:

<https://support.nagios.com/kb/article/nagios-xi-uploading-and-managing-mibs-852.html>

How to Use SNMP Traps with NXTI in Nagios XI 5

Import Existing Trap Definitions

Existing users of Nagios XI before the 5.5 release may already have traps configured in the `/etc/snmp/snmpd.conf` file. These can be imported into NXTI using a script which can save re-inventing the wheel.

After they are imported, the original `snmpd.conf` file needs to be truncated so there are no duplicate trap definitions. Follow these steps to import the `snmpd.conf` file into NXTI.

1. Establish a terminal session to your Nagios XI server as the root user. The first step is to change into the directory and create a backup of the `snmpd.conf` file by executing the following commands:

```
cd /etc/snmp/  
cp snmpd.conf snmpd.conf.original
```

2. Now execute the following command to import the traps:

```
/usr/local/nagiosxi/scripts/nxti_import.php snmpd.conf
```

3. The script will output one line per trap it imports, here is some example output:

```
coldStart  
warmStart  
linkDown  
linkUp  
authenticationFailure
```

4. Now you need to truncate the `snmpd.conf` file by executing the following command:

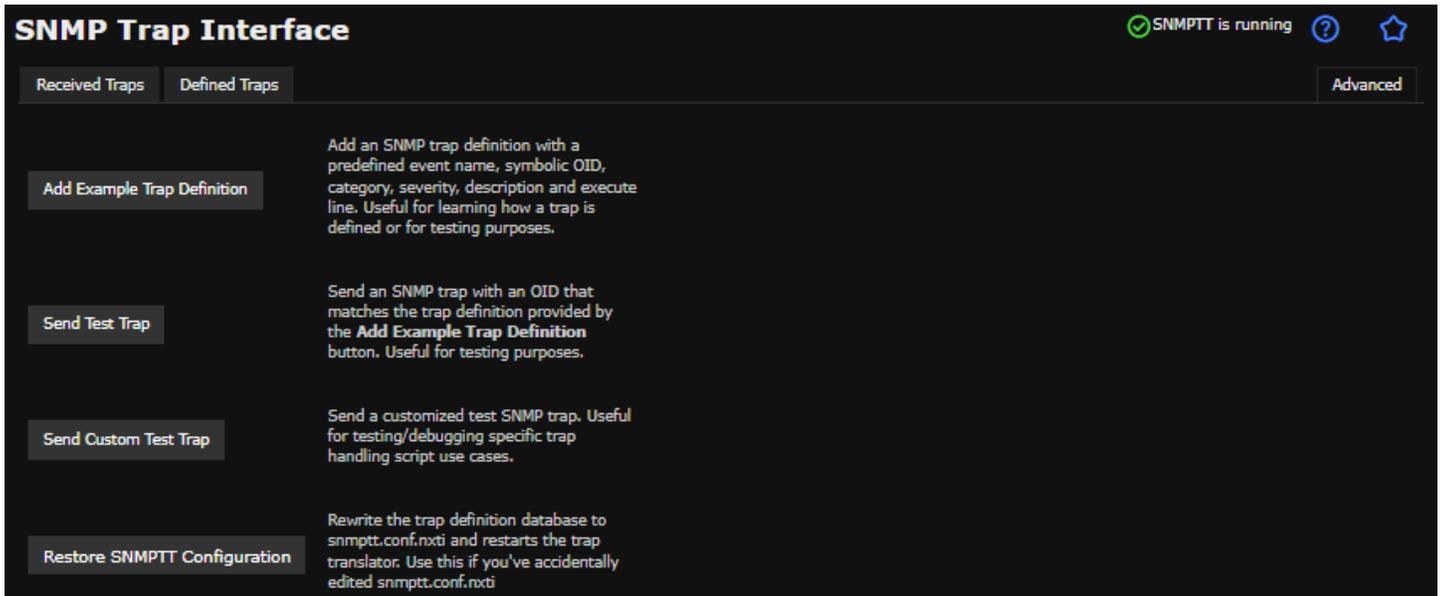
```
echo '' > snmpd.conf
```

5. You can close the terminal session as you have completed this part of the import. Next you need to open NXTI and check to make sure the traps have been imported; you'll find these under the **Defined Traps** tab.
6. The final step is to force the NXTI defined traps into the `snmpd.conf.nxti` file and restart the `snmpd` service. Both steps are performed by navigating to the **Advanced** tab and clicking the **Restore SNMPD Configuration** button.

After completing these steps, you will have successfully imported the existing traps in the `snmpd.conf` file into NXTI.

How to Use SNMP Traps with NXTI in Nagios XI 5

Advanced Features



The **Advanced** tab in NXTI provides various functionalities as explained below.

- Once Click Systems Test
 - Performs a complete test of your local SNMPTT setup. It will add a trap definition, verify its existence in the database, send a matching `snmptrap` command, verify that the trap was received, check the output of the `EXEC` line, and then delete the definition, trap log, and output file.
- Add Example Trap Definition
 - Add an SNMP trap definition that is useful for learning or testing.
- Send Test Trap
 - Allows you to send a trap that matches the example trap definition above.
- Send Custom Test Trap
 - Allows you to send a custom test trap, a modal window appears with fields you can populate.
- Restore SNMPTT Configuration
 - Pushes the settings from the NXTI configuration database back into the `/etc/snmp/snmptt.conf.nxti` file, useful if the file was accidentally edited.
- Show Test File Contents

How to Use SNMP Traps with NXTI in Nagios XI 5

- Displays a modal window with the contents of the `/usr/local/nagiosxi/var/NXTI_Write_Test` file.
- Show Unknown Trap Log
 - Displays a modal window with the contents of the `/var/log/snmpd/snmpdunknown.log` file.
 - This file can be useful for identifying received traps that yet do not have trap definitions created.

Finishing Up

This completes the documentation on **How to Use SNMP Traps with NXTI in Nagios XI**. If you have additional questions or other support-related questions, please visit us at our Nagios Support Forum, Nagios Knowledge Base, or Nagios Library:

[Visit Nagios Support Forum](#)

[Visit Nagios Knowledge Base](#)

[Visit Nagios Library](#)