# Configuring SNMP

This chapter contains the following sections:


- About SNMP, on page 1
- Guidelines and Limitations for SNMP, on page 5
- Default SNMP Settings, on page 5
- Configuring SNMP, on page 6
- Configuring the SNMP Local Engine ID, on page 18
- Disabling SNMP, on page 19
- Verifying the SNMP Configuration, on page 20


# About SNMP

The Simple Network Management Protocol (SNMP) is an application-layer protocol that provides a message format for communication between SNMP managers and agents. SNMP provides a standardized framework and a common language used for the monitoring and management of devices in a network.

## SNMP Functional Overview

The SNMP framework consists of three parts:

- An SNMP manager—The system used to control and monitor the activities of network devices using SNMP.
- An SNMP agent—The software component within the managed device that maintains the data for the device and reports these data, as needed, to managing systems. The Cisco Nexus device supports the agent and MIB. To enable the SNMP agent, you must define the relationship between the manager and the agent.
- A managed information base (MIB)—The collection of managed objects on the SNMP agent

**Note** Cisco Nexus device does not support SNMP sets for Ethernet MIBs.

The Cisco Nexus device supports SNMPv1, SNMPv2c, and SNMPv3. Both SNMPv1 and SNMPv2c use a community-based form of security.

SNMP is defined in RFC 3410 (http://tools.ietf.org/html/rfc3410), RFC 3411 (http://tools.ietf.org/html/rfc3411), RFC 3412 (http://tools.ietf.org/html/rfc3412), RFC 3413 (http://tools.ietf.org/html/rfc3413), RFC 3414 (http://tools.ietf.org/html/rfc3414), RFC 3415 (http://tools.ietf.org/html/rfc3415), RFC 3416 (http://tools.ietf.org/html/rfc3416), RFC 3417 (http://tools.ietf.org/html/rfc3417), RFC 3418 (http://tools.ietf.org/html/rfc3418), and RFC 3584 (http://tools.ietf.org/html/rfc3584).

# SNMP Notifications

A key feature of SNMP is the ability to generate notifications from an SNMP agent. These notifications do not require that requests be sent from the SNMP manager. Notifications can indicate improper user authentication, restarts, the closing of a connection, loss of connection to a neighbor router, or other significant events.

Cisco NX-OS generates SNMP notifications as either traps or informs. A trap is an asynchronous, unacknowledged message sent from the agent to the SNMP managers listed in the host receiver table. Informs are asynchronous messages sent from the SNMP agent to the SNMP manager which the manager must acknowledge receipt of.

Traps are less reliable than informs because the SNMP manager does not send any acknowledgment when it receives a trap. The switch cannot determine if the trap was received. An SNMP manager that receives an inform request acknowledges the message with an SNMP response protocol data unit (PDU). If the Cisco Nexus device never receives a response, it can send the inform request again.

You can configure Cisco NX-OS to send notifications to multiple host receivers.

# SNMPv3

SNMPv3 provides secure access to devices by a combination of authenticating and encrypting frames over the network. The security features provided in SNMPv3 are the following:

- Message integrity—Ensures that a packet has not been tampered with in-transit.
- Authentication—Determines the message is from a valid source.
- Encryption—Scrambles the packet contents to prevent it from being seen by unauthorized sources.

SNMPv3 provides for both security models and security levels. A security model is an authentication strategy that is set up for a user and the role in which the user resides. A security level is the permitted level of security within a security model. A combination of a security model and a security level determines which security mechanism is employed when handling an SNMP packet.

## Security Models and Levels for SNMPv1, v2, and v3

The security level determines if an SNMP message needs to be protected from disclosure and if the message needs to be authenticated. The various security levels that exist within a security model are as follows:

- noAuthNoPriv—Security level that does not provide authentication or encryption. This level is not supported for SNMPv3.
- authNoPriv—Security level that provides authentication but does not provide encryption.
- authPriv—Security level that provides both authentication and encryption.

Three security models are available: SNMPv1, SNMPv2c, and SNMPv3. The security model combined with the security level determine the security mechanism applied when the SNMP message is processed.

***Table 1: SNMP Security Models and Levels***

| Model | Level | Authentication | Encryption | What Happens |
|---|---|---|---|---|
| v1 | noAuthNoPriv | Community string | No | Uses a community string match for authentication. |
| v2c | noAuthNoPriv | Community string | No | Uses a community string match for authentication. |
| v3 | authNoPriv | HMAC-MD5 or HMAC-SHA | No | Provides authentication based on the Hash-Based Message Authentication Code (HMAC) Message Digest 5 (MD5) algorithm or the HMAC Secure Hash Algorithm (SHA). |
| v3 | authPriv | HMAC-MD5 or HMAC-SHA | DES | Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides Data Encryption Standard (DES) 56-bit encryption in addition to authentication based on the Cipher Block Chaning (CBC) DES (DES-56) standard. |

## User-Based Security Model

SNMPv3 User-Based Security Model (USM) refers to SNMP message-level security and offers the following services:

- Message integrity—Ensures that messages have not been altered or destroyed in an unauthorized manner and that data sequences have not been altered to an extent greater than can occur nonmaliciously.
- Message origin authentication—Confirms that the claimed identity of the user who received the data was originated.
- Message confidentiality—Ensures that information is not made available or disclosed to unauthorized individuals, entities, or processes.

SNMPv3 authorizes management operations only by configured users and encrypts SNMP messages.

Cisco NX-OS uses two authentication protocols for SNMPv3:

- HMAC-MD5-96 authentication protocol
- HMAC-SHA-96 authentication protocol

Cisco NX-OS uses Advanced Encryption Standard (AES) as one of the privacy protocols for SNMPv3 message encryption and conforms with RFC 3826.

The **priv** option offers a choice of DES or 128-bit AES encryption for SNMP security encryption. The **priv** option and the **aes-128** token indicates that this privacy password is for generating a 128-bit AES key #.The AES priv password can have a minimum of eight characters. If the passphrases are specified in clear text, you can specify a maximum of 64 characters. If you use the localized key, you can specify a maximum of 130 characters.

**Note** For an SNMPv3 operation using the external AAA server, you must use AES for the privacy protocol in user configuration on the external AAA server.

## CLI and SNMP User Synchronization

SNMPv3 user management can be centralized at the Access Authentication and Accounting (AAA) server level. This centralized user management allows the SNMP agent in Cisco NX-OS to leverage the user authentication service of the AAA server. Once user authentication is verified, the SNMP PDUs are processed further. Additionally, the AAA server is also used to store user group names. SNMP uses the group names to apply the access/role policy that is locally available in the switch.

Any configuration changes made to the user group, role, or password results in database synchronization for both SNMP and AAA.

Cisco NX-OS synchronizes user configuration in the following ways:

- The **auth** passphrase specified in the **snmp-server user** command becomes the password for the CLI user.
- The password specified in the **username** command becomes the **auth** and **priv** passphrases for the SNMP user.
- If you create or delete a user using either SNMP or the CLI, the user is created or deleted for both SNMP and the CLI.
- User-role mapping changes are synchronized in SNMP and the CLI.
- Role changes (deletions or modifications from the CLI) are synchronized to SNMP.

**Note** When you configure passphrase/password in localized key/encrypted format, Cisco NX-OS does not synchronize the user information (passwords, rules, etc.).

## Group-Based SNMP Access

**Note** Because a group is a standard SNMP term used industry-wide, roles are referred to as groups in this SNMP section.

SNMP access rights are organized by groups. Each group in SNMP is similar to a role through the CLI. Each group is defined with three accesses: read access, write access, and notification access. Each access can be enabled or disabled within each group.

You can begin communicating with the agent once your username is created, your roles are set up by your administrator, and you are added to the roles.

# Guidelines and Limitations for SNMP

SNMP has the following configuration guidelines and limitations:

- Commands configured using SNMP SET should be deleted using SNMP SET only. Commands configured using Command Line Interface(CLI) or NX-API should be deleted using CLI or NX-API only.
- Access control list (ACLs) can be applied only to local SNMPv3 users configured on the switch. ACLs cannot be applied to remote SNMPv3 users stored on Authentication, Authorization, and Accounting (AAA) servers.
- Cisco NX-OS supports read-only access to Ethernet MIBs. For more information, see the Cisco NX-OS MIB support list at the following URL ftp://ftp.cisco.com/pub/mibs/supportlists/nexus3000/Nexus3000MIBSupportList.html.
- Cisco NX-OS does not support the SNMPv3 noAuthNoPriv security level.
- Commands configured using SNMP SET should be deleted using SNMP SET only. Commands configured using Command Line Interface(CLI) or NX-API should be deleted using CLI or NX-API only.

# Default SNMP Settings

*Table 2: Default SNMP Parameters*

| Parameters | Default |
|---|---|
| license notifications | Enabled |
| linkUp/Down notification type | ietf-extended |

# Configuring SNMP

## Configuring the SNMP Source Interface

You can configure SNMP to use a specific interface.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **snmp-server source-interface** {**inform** | **trap**} *type slot/port*
3. switch(config)# **show snmp source-interface**

### DETAILED STEPS

#### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **configure terminal** | Enters global configuration mode. |
| **Step 2** | switch(config)# **snmp-server source-interface** {**inform** \| **trap**} *type slot/port* | Configures the source interface for all SNMP packets. The following list contains the valid values for *interface*.<br>• ethernet<br>• loopback<br>• mgmt<br>• port-channel<br>• vlan |
| **Step 3** | switch(config)# **show snmp source-interface** | Displays the configured SNMP source interface. |

#### Example

This example shows how to configure the SNMP source interface:

```
switch(config)# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# snmp-server source-interface inform ethernet 1/10
switch(config)# snmp-server source-interface trap ethernet 1/10
switch(config)# show snmp source-interface
-----------------------------------------------------------------------
Notification                    source-interface
-----------------------------------------------------------------------
trap                              Ethernet1/10
inform                              Ethernet1/10
-----------------------------------------------------------------------
```

# Configuring SNMP Users

**Note** The commands used to configure SNMP users in Cisco NX-OS are different from those used to configure users in Cisco IOS.

### SUMMARY STEPS

1. **configure terminal**
2. switch(config)# **snmp-server user** *name* [**auth** {**md5** | **sha**} *passphrase* [**auto**] [**priv** [**aes-128**] *passphrase*] [**engineID** *id*] [**localizedkey**]]
3. (Optional) switch# **show snmp user**
4. (Optional) **copy running-config startup-config**

### DETAILED STEPS

#### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | switch(config)# **snmp-server user** *name* [**auth** {**md5** \| **sha**} *passphrase* [**auto**] [**priv** [**aes-128**] *passphrase*] [**engineID** *id*] [**localizedkey**]]<br>**Example:**<br>`switch(config)# snmp-server user Admin auth sha abcd1234 priv abcdefgh` | Configures an SNMP user with authentication and privacy parameters.<br>The passphrase can be any case-sensitive, alphanumeric string up to 64 characters.<br>If you use the **localizedkey** keyword, the passphrase can be any case-sensitive, alphanumeric string up to 130 characters.<br>The engineID format is a 12-digit, colon-separated decimal number. |
| **Step 3** | (Optional) switch# **show snmp user**<br>**Example:**<br>`switch(config) # show snmp user` | Displays information about one or more SNMP users. |
| **Step 4** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration. |

### Example

The following example shows how to configure an SNMP user:

```
switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# snmp-server user Admin auth sha abcd1234 priv abcdefgh
```

# Enforcing SNMP Message Encryption

You can configure SNMP to require authentication or encryption for incoming requests. By default, the SNMP agent accepts SNMPv3 messages without authentication and encryption. When you enforce privacy, Cisco NX-OS responds with an authorization error for any SNMPv3 PDU request that uses a security level parameter of either **noAuthNoPriv** or **authNoPriv**.

Use the following command in global configuration mode to enforce SNMP message encryption for a specific user:

| Command | Purpose |
|---|---|
| switch(config)# **snmp-server user** *name* **enforcePriv** | Enforces SNMP message encryption for this user. |

Use the following command in global configuration mode to enforce SNMP message encryption for all users:

| Command | Purpose |
|---|---|
| switch(config)# **snmp-server globalEnforcePriv** | Enforces SNMP message encryption for all users. |

# Assigning SNMPv3 Users to Multiple Roles

After you configure an SNMP user, you can assign multiple roles for the user.

**Note** Only users who belong to a network-admin role can assign roles to other users.

| Command | Purpose |
|---|---|
| switch(config)# **snmp-server user** *name group* | Associates this SNMP user with the configured user role. |

# Creating SNMP Communities

You can create SNMP communities for SNMPv1 or SNMPv2c.

| Command | Purpose |
|---|---|
| switch(config)# **snmp-server community** *name group* {**ro** \| **rw**} | Creates an SNMP community string. |

# Filtering SNMP Requests

You can assign an access list (ACL) to a community to filter incoming SNMP requests. If the assigned ACL allows the incoming request packet, SNMP processes the request. If the ACL denies the request, SNMP drops the request and sends a system message.

Create the ACL with the following parameters:

- Source IP address
- Destination IP address
- Source port
- Destination port
- Protocol (UDP or TCP)

The ACL applies to both IPv4 and IPv6 over UDP and TCP. After creating the ACL, assign the ACL to the SNMP community.

**Tip** For more information about creating ACLs, see the NX-OS security configuration guide for the Cisco Nexus Series software that you are using.

Use the following command in global configuration mode to assign an ACL to a community to filter SNMP requests:

| Command | Purpose |
|---|---|
| switch(config)# **snmp-server community** *community name* **use-acl** *acl-name*<br>**Example:**<br>`switch(config)# snmp-server community public use-acl my_acl_for_public` | Assigns an IPv4 or IPv6 ACL to an SNMP community to filter SNMP requests. |

# Configuring SNMP Notification Receivers

You can configure Cisco NX-OS to generate SNMP notifications to multiple host receivers.

You can configure a host receiver for SNMPv1 traps in a global configuration mode.

| Command | Purpose |
|---|---|
| switch(config)# **snmp-server host** *ip-address* **traps version 1** *community* [**udp_port** *number*] | Configures a host receiver for SNMPv1 traps. The *ip-address* can be an IPv4 or IPv6 address. The community can be any alphanumeric string up to 255 characters. The UDP port number range is from 0 to 65535. |

You can configure a host receiver for SNMPv2c traps or informs in a global configuration mode.

| Command | Purpose |
|---|---|
| switch(config)# **snmp-server host** *ip-address* {**traps** \| **informs**} **version 2c** *community* [**udp_port** *number*] | Configures a host receiver for SNMPv2c traps or informs. The *ip-address* can be an IPv4 or IPv6 address. The community can be any alphanumeric string up to 255 characters. The UDP port number range is from 0 to 65535. |

You can configure a host receiver for SNMPv3 traps or informs in a global configuration mode.

| Command | Purpose |
|---|---|
| switch(config)# **snmp-server host** *ip-address* {**traps** \| **informs**} **version 3** {**auth** \| **noauth** \| **priv**} *username* [**udp_port** *number*] | Configures a host receiver for SNMPv2c traps or informs. The *ip-address* can be an IPv4 or IPv6 address. The username can be any alphanumeric string up to 255 characters. The UDP port number range is from 0 to 65535. |

**Note** The SNMP manager must know the user credentials (authKey/PrivKey) based on the SNMP engineID of the Cisco Nexus device to authenticate and decrypt the SNMPv3 messages.

The following example shows how to configure a host receiver for an SNMPv1 trap:

```
switch(config)# snmp-server host 192.0.2.1 traps version 1 public
```

The following example shows how to configure a host receiver for an SNMPv2 inform:

```
switch(config)# snmp-server host 192.0.2.1 informs version 2c public
```

The following example shows how to configure a host receiver for an SNMPv3 inform:

```
switch(config)# snmp-server host 192.0.2.1 informs version 3 auth NMS
```

# Configuring SNMP Notification Receivers with VRFs

You can configure Cisco NX-OS to use a configured VRF to reach the host receiver. SNMP adds entries into the cExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MIB when you configure the VRF reachability and filtering options for an SNMP notification receiver.

**Note** You must configure the host before configuring the VRF reachability or filtering options.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch# **snmp-server host** *ip-address* **use-vrf** *vrf_name* [**udp_port** *number*]
3. (Optional) switch(config)# **copy running-config startup-config**

### DETAILED STEPS

#### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **configure terminal** | Enters global configuration mode. |
| **Step 2** | switch# **snmp-server host** *ip-address* **use-vrf** *vrf_name* [**udp_port** *number*] | Configures SNMP to use the selected VRF to communicate with the host receiver. The IP address can be an IPv4 or IPv6 address. The VRF name can be any alphanumeric string up to 255 characters. The UDP port number range is from 0 to 65535. This command adds an entry into thc ExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MB. |
| **Step 3** | (Optional) switch(config)# **copy running-config startup-config** | Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration. |

#### Example

The following example shows how to configure the SNMP server host with IP address 192.0.2.1 to use the VRF named "Blue:"

```
switch# configuration terminal
switch(config)# snmp-server host 192.0.2.1 use-vrf Blue
switch(config)# copy running-config startup-config
```

# Filtering SNMP Notifications Based on a VRF

You can configure Cisco NX-OS filter notifications based on the VRF in which the notification occurred.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **snmp-server host** *ip-address* **filter-vrf** *vrf_name* [**udp_port** *number*]
3. (Optional) switch(config)# **copy running-config startup-config**

### DETAILED STEPS

#### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **configure terminal** | Enters global configuration mode. |
| **Step 2** | switch(config)# **snmp-server host** *ip-address* **filter-vrf** *vrf_name* [**udp_port** *number*] | Filters notifications to the notification host receiver based on the configured VRF. The IP address can be an IPv4 or IPv6 address. The VRF name can be any alphanumeric |

| | Command or Action | Purpose |
|---|---|---|
| | | string up to 255 characters. The UDP port number range is from 0 to 65535.<br>This command adds an entry into thc ExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MB. |
| **Step 3** | (Optional) switch(config)# **copy running-config startup-config** | Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration. |

### Example

The following example shows how to configure filtering of SNMP notifications based on a VRF:

```
switch# configuration terminal
switch(config)# snmp-server host 192.0.2.1 filter-vrf Red
switch(config)# copy running-config startup-config
```

# Configuring SNMP for Inband Access

You can configure SNMP for inband access using the following:

- Using SNMP v2 without context—You can use a community that is mapped to a context. In this case, the SNMP client does not need to know about the context.
- Using SNMP v2 with context—The SNMP client needs to specify the context by specifying a community; for example, <community>@<context>.
- Using SNMP v3—You can specify the context.

### SUMMARY STEPS

1. switch# **configuration terminal**
2. switch(config)# **snmp-server context** *context-name* **vrf** *vrf-name*
3. switch(config)# **snmp-server community** *community-name* **group** *group-name*
4. switch(config)# **snmp-server mib community-map** *community-name* **context** *context-name*

### DETAILED STEPS

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **configuration terminal** | Enters global configuration mode. |
| **Step 2** | switch(config)# **snmp-server context** *context-name* **vrf** *vrf-name* | Maps an SNMP context to the management VRF or default VRF. Custom VRFs are not supported. |

| | Command or Action | Purpose |
|---|---|---|
| | | The names can be any alphanumeric string up to 32 characters. |
| **Step 3** | switch(config)# **snmp-server community** *community-name* **group** *group-name* | Maps an SNMPv2c community to an SNMP context and identifies the group to which the community belongs. The names can be any alphanumeric string up to 32 characters. |
| **Step 4** | switch(config)# **snmp-server mib community-map** *community-name* **context** *context-name* | Maps an SNMPv2c community to an SNMP context. The names can be any alphanumeric string up to 32 characters. |

### Example

The following SNMPv2 example shows how to map a community named snmpdefault to a context:

```
switch# config t
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# snmp-server context def vrf default
switch(config)# snmp-server community snmpdefault group network-admin
switch(config)# snmp-server mib community-map snmpdefault context def
switch(config)#
```

The following SNMPv2 example shows how to configure and inband access to the community comm which is not mapped:

```
switch# config t
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# snmp-server context def vrf default
switch(config)# snmp-server community comm group network-admin
switch(config)#
```

The following SNMPv3 example shows how to use a v3 username and password:

```
switch# config t
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# snmp-server context def vrf default
switch(config)#
```

# Enabling SNMP Notifications

You can enable or disable notifications. If you do not specify a notification name, Cisco NX-OS enables all notifications.

**Note** The **snmp-server enable traps** CLI command enables both traps and informs, depending on the configured notification host receivers.

The following table lists the CLI commands that enable the notifications for Cisco NX-OS MIBs.

***Table 3: Enabling SNMP Notifications***

| MIB | Related Commands |
|---|---|
| All notifications | **snmp-server enable traps** |

| MIB | Related Commands |
|---|---|
| CISCO-ERR-DISABLE-MIB | **snmp-server enable traps show interface status** |
| Q-BRIDGE-MIB | **snmp-server enable traps show mac address-table** |
| CISCO-SWITCH-QOS-MIB | **snmp-server enable traps show hardware internal buffer info pkt-stats** |
| BRIDGE-MIB | **snmp-server enable traps bridge newroot**<br>**snmp-server enable traps bridge topologychange** |
| CISCO-AAA-SERVER-MIB | **snmp-server enable traps aaa** |
| ENITY-MIB,<br>CISCO-ENTITY-FRU-CONTROL-MIB,<br>CISCO-ENTITY-SENSOR-MIB | **snmp-server enable traps entity**<br>**snmp-server enable traps entity fru** |
| CISCO-LICENSE-MGR-MIB | **snmp-server enable traps license** |
| IF-MIB | **snmp-server enable traps link** |
| CISCO-PSM-MIB | **snmp-server enable traps port-security** |
| SNMPv2-MIB | **snmp-server enable traps snmp**<br>**snmp-server enable traps snmp authentication** |
| CISCO-FCC-MIB | **snmp-server enable traps fcc** |
| CISCO-DM-MIB | **snmp-server enable traps fcdomain** |
| CISCO-NS-MIB | **snmp-server enable traps fcns** |
| CISCO-FCS-MIB | **snmp-server enable traps fcs discovery-complete**<br>**snmp-server enable traps fcs request-reject** |
| CISCO-FDMI-MIB | **snmp-server enable traps fdmi** |
| CISCO-FSPF-MIB | **snmp-server enable traps fspf** |
| CISCO-PSM-MIB | **snmp-server enable traps port-security** |
| CISCO-RSCN-MIB | **snmp-server enable traps rscn**<br>**snmp-server enable traps rscn els**<br>**snmp-server enable traps rscn ils** |

| MIB | Related Commands |
|---|---|
| CISCO-ZS-MIB | **snmp-server enable traps zone**<br>**snmp-server enable traps zone default-zone-behavior-change**<br>**snmp-server enable traps zone enhanced-zone-db-change**<br>**snmp-server enable traps zone merge-failure**<br>**snmp-server enable traps zone merge-success**<br>**snmp-server enable traps zone request-reject**<br>**snmp-server enable traps zone unsupp-mem** |
| CISCO-CONFIG-MAN-MIB<br>**Note**<br>Supports no MIB objects except the following notification: ccmCLIRunningConfigChanged | **snmp-server enable traps config** |

**Note** The license notifications are enabled by default.

To enable the specified notification in the global configuration mode, perform one of the following tasks:

| Command | Purpose |
|---|---|
| switch(config)# **snmp-server enable traps** | Enables all SNMP notifications. |
| switch(config)# **snmp-server enable traps aaa** [**server-state-change**] | Enables the AAA SNMP notifications. |
| switch(config)# **snmp-server enable traps entity** [**fru**] | Enables the ENTITY-MIB SNMP notifications. |
| switch(config)# **snmp-server enable traps license** | Enables the license SNMP notification. |
| switch(config)# **snmp-server enable traps port-security** | Enables the port security SNMP notifications. |
| switch(config)# **snmp-server enable traps snmp** [**authentication**] | Enables the SNMP agent notifications. |

# Configuring Link Notifications

You can configure which linkUp/linkDown notifications to enable on a device. You can enable the following types of linkUp/linkDown notifications:

- cieLinkDown—Enables the Cisco extended link state down notification.
- cieLinkUp—Enables the Cisco extended link state up notification.
- cisco-xcvr-mon-status-chg—Enables the Cisco interface transceiver monitor status change notification.
- delayed-link-state-change—Enables the delayed link state change.

- extended-linkUp—Enables the Internet Engineering Task Force (IETF) extended link state up notification.
- extended-linkDown—Enables the IETF extended link state down notification.
- linkDown—Enables the IETF Link state down notification.
- linkUp—Enables the IETF Link state up notification.

### SUMMARY STEPS

1. **configure terminal**
2. **snmp-server enable traps link** [**cieLinkDown** | **cieLinkUp** | **cisco-xcvr-mon-status-chg** | **delayed-link-state-change**] | **extended-linkUp** | **extended-linkDown** | **linkDown** | **linkUp**]

### DETAILED STEPS

#### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **snmp-server enable traps link** [**cieLinkDown** | **cieLinkUp** | **cisco-xcvr-mon-status-chg** | **delayed-link-state-change**] | **extended-linkUp** | **extended-linkDown** | **linkDown** | **linkUp**]<br>**Example:**<br>`switch(config)# snmp-server enable traps link cieLinkDown` | Enables the link SNMP notifications. |

# Disabling Link Notifications on an Interface

You can disable linkUp and linkDown notifications on an individual interface. You can use these limit notifications on a flapping interface (an interface that transitions between up and down repeatedly).

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot*/*port*
3. switch(config -if)# **no snmp trap link-status**

### DETAILED STEPS

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **configure terminal** | Enters global configuration mode. |
| **Step 2** | switch(config)# **interface** *type slot/port* | Specifies the interface to be changed. |
| **Step 3** | switch(config -if)# **no snmp trap link-status** | Disables SNMP link-state traps for the interface. This feature is enabled by default. |

# Enabling One-Time Authentication for SNMP over TCP

You can enable a one-time authentication for SNMP over a TCP session.

| Command | Purpose |
|---|---|
| switch(config)# **snmp-server tcp-session** [**auth**] | Enables a one-time authentication for SNMP over a TCP session. This feature is disabled by default. |

# Assigning SNMP Switch Contact and Location Information

You can assign the switch contact information, which is limited to 32 characters (without spaces), and the switch location.

### SUMMARY STEPS

1. switch# **configuration terminal**
2. switch(config)# **snmp-server contact** *name*
3. switch(config)# **snmp-server location** *name*
4. (Optional) switch# **show snmp**
5. (Optional) switch# **copy running-config startup-config**

### DETAILED STEPS

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **configuration terminal** | Enters global configuration mode. |
| **Step 2** | switch(config)# **snmp-server contact** *name* | Configures sysContact, the SNMP contact name. |
| **Step 3** | switch(config)# **snmp-server location** *name* | Configures sysLocation, the SNMP location. |
| **Step 4** | (Optional) switch# **show snmp** | Displays information about one or more destination profiles. |
| **Step 5** | (Optional) switch# **copy running-config startup-config** | Saves this configuration change. |

# Configuring the Context to Network Entity Mapping

You can configure an SNMP context to map to a logical network entity, such as a protocol instance or VRF.

### SUMMARY STEPS

1. switch# **configuration terminal**
2. switch(config)# **snmp-server context** *context-name* [**instance** *instance-name*] [**vrf** *vrf-name*] [**topology** *topology-name*]
3. switch(config)# **snmp-server mib community-map** *community-name* **context** *context-name*
4. (Optional) switch(config)# **no snmp-server context** *context-name* [**instance** *instance-name*] [**vrf** *vrf-name*] [**topology** *topology-name*]

### DETAILED STEPS

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **configuration terminal** | Enters global configuration mode. |
| **Step 2** | switch(config)# **snmp-server context** *context-name* [**instance** *instance-name*] [**vrf** *vrf-name*] [**topology** *topology-name*] | Maps an SNMP context to a protocol instance, VRF, or topology. The names can be any alphanumeric string up to 32 characters. |
| **Step 3** | switch(config)# **snmp-server mib community-map** *community-name* **context** *context-name* | Maps an SNMPv2c community to an SNMP context. The names can be any alphanumeric string up to 32 characters. |
| **Step 4** | (Optional) switch(config)# **no snmp-server context** *context-name* [**instance** *instance-name*] [**vrf** *vrf-name*] [**topology** *topology-name*] | Deletes the mapping between an SNMP context and a protocol instance, VRF, or topology. The names can be any alphanumeric string up to 32 characters.<br>**Note**<br>Do not enter an instance, VRF, or topology to delete a context mapping. If you use the **instance**, **vrf**, or **topology** keywords, you configure a mapping between the context and a zero-length string. |

# Configuring the SNMP Local Engine ID

Beginning with Cisco NX-OS Release 7.0(3)F3(1), you can configure the engine ID on a local device.

### SUMMARY STEPS

1. **configure terminal**
2. **snmp-server engineID local** *engineid-string*
3. **show snmp engineID**
4. **[no] snmp-server engineID local** *engineid-string*
5. **copy running-config startup-config**

### DETAILED STEPS

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **snmp-server engineID local** *engineid-string*<br>**Example:**<br>`switch(config)# snmp-server engineID local AA:BB:CC:1A:2C:10` | Changes the SNMP engineID of the local device.<br>The local engine ID should be configured as a list of colon-specified hexadecimal octets, where there are even number of hexadecimal characters that range from 10 to 64 and every two hexadecimal characters are separated by a colon. For example, i80:00:02:b8:04:61:62:63. |
| **Step 3** | **show snmp engineID**<br>**Example:**<br>`switch(config)# show snmp engineID` | Displays the identification of the configured SNMP engine. |
| **Step 4** | **[no] snmp-server engineID local** *engineid-string*<br>**Example:**<br>`switch(config)# no snmp-server engineID local AA:BB:CC:1A:2C:10` | Disables the local engine ID and the default auto-generated engine ID is configured. |
| **Step 5** | Required: **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |

# Disabling SNMP

### SUMMARY STEPS

1. **configure terminal**
2. switch(config) # **no snmp-server protocol enable**

**DETAILED STEPS**

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | switch(config) # **no snmp-server protocol enable**<br>**Example:**<br>`no snmp-server protocol enable` | Disables SNMP.<br>SNMP is disabled by default. |

# Verifying the SNMP Configuration

To display SNMP configuration information, perform one of the following tasks:

| Command | Purpose |
|---|---|
| **show snmp** | Displays the SNMP status. |
| **show snmp community** | Displays the SNMP community strings. |
| **show interface snmp-ifindex** | Displays the SNMP ifIndex value for all interfaces (from IF-MIB). |
| **show running-config snmp [all]** | Displays the SNMP running configuration. |
| **show snmp engineID** | Displays the SNMP engineID. |
| **show snmp group** | Displays SNMP roles. |
| **show snmp sessions** | Displays SNMP sessions. |
| **show snmp context** | Displays the SNMP context mapping. |
| **show snmp host** | Displays information about configured SNMP hosts. |
| **show snmp source-interface** | Displays information about configured source interfaces. |
| **show snmp trap** | Displays the SNMP notifications enabled or disabled. |
| **show snmp user** | Displays SNMPv3 users. |