

Simple Network Management Protocol

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.

Note The license notifications are enabled by default.

- SNMP Fundamentals, on page 1
- Guidelines and Limitations, on page 13
- Default Settings for SNMP, on page 16
- Configure SNMP, on page 16
- Configure SNMP Local Engine ID, on page 40
- Verify SNMP Configuration, on page 41
- Configuration Examples, on page 42
- Additional References, on page 43

SNMP Fundamentals

Framework

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 SNMP is defined in RFCs 3411 to 3418. The device supports SNMPv1, SNMPv2c, and SNMPv3. Both SNMPv1 and SNMPv2c use a community-based form of security.

Cisco NX-OS supports SNMP over IPv6.

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.

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 device 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 device never receives a response, it can send the inform request again.

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

The following table lists the SNMP traps that are enabled by default.

Trap Type	Description
generic	: coldStart
entity	: entity_fan_status_change
entity	: entity_mib_change
entity	: entity_module_status_change
entity	: entity_module_inserted
entity	: entity_module_removed
entity	: entity_power_out_change
entity	: entity_power_status_change
entity	: entity_unrecognised_module
link	: cErrDisableInterfaceEventRev1
link	: cieLinkDown
link	: cieLinkUp
link	: cmn-mac-move-notification
link	: delayed-link-state-change
link	: extended-linkDown
link	: extended-linkUp

Trap Type	Description
link	: linkDown
link	: linkUp
rf	: redundancy_framework
upgrade	: UpgradeOpNotifyOnCompletion
upgrade	: UpgradeJobStatusNotify
entity	: entity_sensor
rmon	: fallingAlarm
rmon	: hcRisingAlarm
rmon	: hcFallingAlarm
rmon	: risingAlarm

Security Levels and Models

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. SNMPv3 provides for both security models and security levels.

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.

The security models available are:

- SNMPv1
- SNMPv2c
- SNMPv3

The following table identifies what the combinations of security models and levels mean.

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.

Model	Level	Authentication	Encryption	What Happens
v2c	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
v3	authNoPriv	HMAC-MD5, HMAC-SHA, or SHA-256	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, HMAC-SHA, or SHA-256	DES or AES-128	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 Chaining (CBC) DES (DES-56) standard.

SNMP Logical Entities

A device can support multiple instances of a logical network entity, such as protocol instances or virtual routing and forwarding (VRF) instances. Most existing MIBs cannot distinguish between the multiple logical network entities. For example, the original OSPF-MIB assumes a single protocol instance on a device, but you can now configure multiple OSPF instances on a device.

Context

An SNMP context is a collection of management information that you can access through the SNMP agent.

Groups

SNMP access rights are organized by groups. Each group in SNMP is similar to a role through the CLI. Each group is defined with read access or read-write access.

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.

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

Embedded Event Manager (EEM)

The Embedded Event Manager (EEM) feature monitors events, including SNMP MIB objects, and triggers an action based on these events. One of the actions could be to send an SNMP notification. EEM sends the cEventMgrPolicyEvent of CISCO-EMBEDDED-EVENT-MGR-MIB as the SNMP notification.

VRF Support

SNMP is also VRF aware. You can configure SNMP to use a particular VRF to reach the SNMP notification host receiver. You can also configure SNMP to filter notifications to an SNMP host receiver based on the VRF where the notification occurred.

Cisco NX-OS Support

Cisco NX-OS supports one instance of the SNMP. Cisco NX-OS supports stateless restarts for SNMP. After a reboot or supervisor switchover, Cisco NX-OS applies the running configuration. Cisco NX-OS supports SNMP over IPv6.

Multiple Instance Support

SNMP supports multiple MIB module instances and maps them to logical network entities. Cisco NX-OS supports the CISCO-CONTEXT-MAPPING-MIB to map between SNMP contexts and logical network entities. You can associate an SNMP context to a VRF, protocol instance, or topology.

SNMPv3 - uses contexts to distinguish between these multiple instances. A device can support multiple contexts for different logical network entities. An SNMP context allows the SNMP manager to access one of the multiple instances of a MIB module supported on the device for the different logical network entities. SNMPv3 supports contexts with the contextName field of the SNMPv3 PDU. You can map this contextName field to a particular protocol instance or VRF.

SNMPv2c - you can map the SNMP community to a context using the snmpCommunityContextName MIB object in the SNMP-COMMUNITY-MIB (RFC 3584). You can then map this snmpCommunityContextName to a particular protocol instance or VRF using the CISCO-CONTEXT-MAPPING-MIB or the CLI.

Note SNMP is defined in RFCs 3411 to 3418. The device supports SNMPv1, SNMPv2c, and SNMPv3. Both SNMPv1 and SNMPv2c use a community-based form of security.

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

SNMP is defined in RFCs 3411 to 3418.

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

Cisco NX-OS supports SNMP over IPv6.

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.

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 device 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 device never receives a response, it can send the inform request again.

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

The following table lists the SNMP traps that are enabled by default.

Trap Type	Description
generic	: coldStart
entity	: entity_fan_status_change
entity	: entity_mib_change
entity	: entity_module_status_change
entity	: entity_module_inserted
entity	: entity_module_removed
entity	: entity_power_out_change
entity	: entity_power_status_change
entity	: entity_unrecognised_module
link	: cErrDisableInterfaceEventRev1
link	: cieLinkDown
link	: cieLinkUp
link	: cmn-mac-move-notification
link	: delayed-link-state-change
link	: extended-linkDown

Trap Type	Description
link	: extended-linkUp
link	: linkDown
link	: linkUp
rf	: redundancy_framework
upgrade	: UpgradeOpNotifyOnCompletion
upgrade	: UpgradeJobStatusNotify
entity	: entity_sensor
rmon	: fallingAlarm
rmon	: hcRisingAlarm
rmon	: hcFallingAlarm
rmon	: risingAlarm

SNMPv3

SNMPv3 provides secure access to devices by a combination of authenticating and encrypting frames over the network. The 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.
- Authentication—Ensures that the claimed identity of the user on whose behalf received data was originated is confirmed.
- Encryption—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.

Authentication Protocols

NX-OS uses three authentication protocols for SNMPv3:

- HMAC-MD5-96 authentication protocol
- HMAC-SHA-96 authentication protocol
- SHA-256 authentication protocol

Note Beginning with Cisco NX-OS release 9.3(7), HMAC-SHA-256 authentication protocol is used for SNMPv3. When SHA-256 SNMP users are configured on the switch, ISSD is recommended by **install all** cmd else there will be config loss.

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 indicate 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 case-sensitive, alphanumeric characters. If you use the localized key, you can specify a maximum of 130 characters.

Note SNMPWALK with remote user will only work if user has been created as SNMP user. DCNM/ND is the only NMS which creates SNMP remote user. None of the remote or external AAA server directly creates SNMP remote user.

User Configuration 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 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.

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

- The authentication passphrase specified in the **snmp-server user** command becomes the password for the CLI user.
- The password specified in the **username** command becomes the authentication and privacy 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 a passphrase/password in localized key/encrypted format, NX-OS does not synchronize the user information (passwords, roles, and so on). Cisco NX-OS holds the synchronized user configuration for 60 minutes by default.

Security Models and Levels for SNMPv1, v2, 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. The following table identifies what the combinations of security models and levels mean.

Table 2: 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, HMAC-SHA, or SHA-256	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, HMAC-SHA, or SHA-256	DES or AES-128	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 Chaining (CBC) DES (DES-56) standard.

User-Based Security Model

The 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—Ensures that the claimed identity of the user on whose behalf received data was originated is confirmed.
- 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.

NX-OS uses three authentication protocols for SNMPv3:

- HMAC-MD5-96 authentication protocol
- HMAC-SHA-96 authentication protocol
- SHA-256 authentication protocol

Beginning with Cisco NX-OS release 9.3(7), HMAC-SHA-256 authentication protocol is used for SNMPv3.



Note When SHA-256 SNMP users are configured on the switch, ISSD is recommended by **install all** cmd else there will be config loss.

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 indicate 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 case-sensitive, alphanumeric 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 the 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 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.

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

- The authentication passphrase specified in the **snmp-server user** command becomes the password for the CLI user.
- The password specified in the **username** command becomes the authentication and privacy 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 a passphrase/password in localized key/encrypted format, NX-OS does not synchronize the user information (passwords, roles, and so on).

Cisco NX-OS holds the synchronized user configuration for 60 minutes by default.

Disable Security and SNMP User Synchronization

Beginning with Cisco NX-OS Release 10.2(2)F, the following desynchronization command is introduced to provide you an option to disable the user synchronization between the SNMP and the security (AAA or CLI) components:

snmp-server disable snmp-aaa sync

You can execute this command from the configure terminal on the Nexus switches. By default, the **no** form of the desynchronization command is available on the switch.

When the no-form of the desynchronization command is enabled on the device, for example, `switch (config)#` **no snmp-server disable snmp-aaa sync**, a user created through **snmp-server user** CLI results in the creation of a **username** CLI for that user in the running configuration and conversely. So, the user can log in to the switch, using the authentication credentials mentioned in the **snmp-server user** CLI or the **username** CLI, at the time of creation/updation, and will also be able to perform SNMP operations from a network manager on the switch. Thus, the **no** form of the desynchronization command ensures that the user synchronization between the SNMP and the AAA functions the way it did in the releases prior to 10.2(2)F.

When the desynchronization command is enabled on the device, for example, `switch (config)#` **snmp-server disable snmp-aaa sync**, a user created through the **snmp-server user** command does not create a username configuration for that user. So, the user cannot log in to the switch and is only allowed to do SNMP operations through a network manager on the switch. Similarly, creation of a security user through the **username** CLI does not create a corresponding **snmp-server user** CLI for the user. This user will be able to log in to the switch but will not be able to perform any SNMP operation on the switch. This is a new feature that the desynchronization command has introduced from Release 10.2(2)F.

You can view the status of the desynchronization command in one of the following ways:

- The value of the field `SNMP-AAA sync disable` in the output of the CLI **show snmp internal globals**
- The value of the field `disableSnmpAaaSync` in the `sys/snmp/inst/globals` MO
- The CLI print in the **show-running-config** output and **show-running-config-snmp** output or **show-running-all** output, based on whether the command is enabled or disabled, respectively

Remote Users

With regard to remote users, who are authenticated for login through external servers using protocols such as RADIUS and TACACS+, when the desynchronization command is enabled on the switch, the remote users

cannot be created in SNMP. For more information, refer to the *Configuring AAA* chapter in the *Cisco Nexus 9000 NX-OS Security Configuration Guide*.

However, when the **no** form of the desynchronization command is enabled on the switch, if a remote user is created in AAA, the corresponding user is created in SNMP as well. Furthermore, the user will not be available in the running-config output of SNMP, but will be able to perform SNMP operations on the managed device, which is an existing feature prior to Release 10.2(2)F. However, external NMS client's originated snmpwalk and so on does not go through unless the **snmp-sync-password** command is executed in the device for the remote user created in AAA.

DCNM Security Users

The security users created using DCNM (also called as Nexus Dashboard Fabric Controller from Release 12.0.1a onwards) will not have a corresponding SNMPv3 profile when the desynchronization command is enabled. When the synchronization is disabled, the users created on the security component can log in to the switch, but the switches will not be discovered by the controller, as the controller uses the SNMP configuration created for the security user to discover the switch. Furthermore, the SNMP does not recognize the security users created due to the desynchronized state of the userDB, resulting in failure to discover the switch. Therefore, to have the switches discovered by the controller, the SNMP user must be explicitly created. It is not recommended to use the desynchronization command along with DCNM functionality. For more information, refer to the *Cisco Nexus 9000 NX-OS Security Configuration Guide*.

ISSD and ISSU

In general, if SNMP user synchronization has been disabled, do not enable SNMP user synchronization unless all the desynchronized users are removed. A running configuration with such a combination will result in a configuration replace failure.

The only way to achieve the desynchronized state in older releases without the desynchronization command is as follows:

- If the Disruptive/ND-ISSD is performed from a desynchronized state to a release without the desynchronization command, the desynchronized databases will be ported as-is through ISSD to the previous release.

Note Any modifications done to the user database after such ISSD will be synchronized between SNMP and security components.

After such ISSD, ISSU to a release with desynchronization command brings in the desynchronized user database as-is, but the desynchronization command comes up in its default **no** form. If required, enable the desynchronization command.

Group-Based SNMP Access

Note Because *group* is a standard SNMP term used industry-wide, we refer to roles 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 read access or read-write access.

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.

SNMP and Embedded Event Manager

The Embedded Event Manager (EEM) feature monitors events, including SNMP MIB objects, and triggers an action based on these events. One of the actions could be to send an SNMP notification. EEM sends the cEventMgrPolicyEvent of CISCO-EMBEDDED-EVENT-MGR-MIB as the SNMP notification.

Multiple Instance Support

A device can support multiple instances of a logical network entity, such as protocol instances or virtual routing and forwarding (VRF) instances. Most existing MIBs cannot distinguish between these multiple logical network entities. For example, the original OSPF-MIB assumes a single protocol instance on a device, but you can now configure multiple OSPF instances on a device.

SNMPv3 uses contexts to distinguish between these multiple instances. An SNMP context is a collection of management information that you can access through the SNMP agent. A device can support multiple contexts for different logical network entities. An SNMP context allows the SNMP manager to access one of the multiple instances of a MIB module supported on the device for the different logical network entities.

Cisco NX-OS supports the CISCO-CONTEXT-MAPPING-MIB to map between SNMP contexts and logical network entities. You can associate an SNMP context to a VRF, protocol instance, or topology.

SNMPv3 supports contexts with the contextName field of the SNMPv3 PDU. You can map this contextName field to a particular protocol instance or VRF.

For SNMPv2c, you can map the SNMP community to a context using the snmpCommunityContextName MIB object in the SNMP-COMMUNITY-MIB (RFC 3584). You can then map this snmpCommunityContextName to a particular protocol instance or VRF using the CISCO-CONTEXT-MAPPING-MIB or the CLI.

High Availability for SNMP

Cisco NX-OS supports stateless restarts for SNMP. After a reboot or supervisor switchover, Cisco NX-OS applies the running configuration.

Virtualization Support for SNMP

Cisco NX-OS supports one instance of the SNMP. SNMP supports multiple MIB module instances and maps them to logical network entities.

SNMP is also VRF aware. You can configure SNMP to use a particular VRF to reach the SNMP notification host receiver. You can also configure SNMP to filter notifications to an SNMP host receiver based on the VRF where the notification occurred.

Guidelines and Limitations

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.
- When you create or edit a user in AAA using clear text password, SNMP creates or edits the user to have default auth (md5) and priv types.

  When you create or edit a user in SNMP using clear text password, AAA creates or edits the user to have default password type (type 5).
- 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.
- Do not enable SNMP user synchronisation after it has been disabled unless all desynchronised users are removed. A running configuration with such a combination will result in a configuration replace failure.
- Cisco NX-OS supports read-only access to some SNMP MIBs. See the Cisco NX-OS MIB support list at the following URL for more information: https://cisco.github.io/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html
- When VLAN 1 is not allowed on a trunk, the global SNMP context might not return `BRIDGE-MIB::dot1dBasePort` and `BRIDGE-MIB::dot1dBasePortIfIndex` mappings for that trunk. To retrieve the mappings, poll `BRIDGE-MIB` in the SNMPv3 context for the relevant VLAN, or use `CISCO-IF-EXTENSION-MIB::cieIfDot1dBaseMappingPort` and reverse the mapping in your network-management application.
- Cisco NX-OS does not support the SNMPv3 noAuthNoPriv security level.
- Cisco Nexus 9000 Series switches and the Cisco Nexus 3164Q, 31128PQ, 3232C, and 3264Q switches support the configuration of the SNMP local engine ID.
- For a nondisruptive downgrade path to an earlier release, if a local engine ID has been configured, then you must unconfigure the local engine ID, and then reconfigure the SNMP users and the community strings.
- Special characters @ and % are not allowed in the SNMP community string.
- The default SNMP PDU value is 1500 bytes. The SNMP agent drops any response PDU that is greater than 1500 bytes, causing the SNMP request to fail. To receive MIB data values larger than 1500 bytes, use the **snmp-server packetsize** <*byte-count*> command to reconfigure the packet size. The valid byte-count range is from 484 to 17382. When a GETBULK response exceeds the packet size, the data can get truncated.
- You must use either the CLI or SNMP to configure a feature on your switch. Do not configure a feature using both interfaces to the switch.
- Using cefcFanTrayOperStatus snmpwalk on an individual fan OID tree where the fan is not populated in chassis, can return a response for next OID entry in the tree. To prevent this behavior, use the -CI option in *snmpwalk*.

  The behavior is not seen when polling parent OID, or when using getmany.
- Cisco Nexus 9000 series switches support upto 20000 flash files for *snmpwalk* request.
- There must be at least one running BGP instance to have full, proper functional behavior of SNMP traps. Configure a BGP routing instance before configuring any snmp-server traps related commands.

- Beginning with Release 10.1(1), AES-128 is the recommended encryption algorithm, as it is a strong encryption algorithm. However, DES encryption is also supported.

  Downgrade: In-Service System Downgrade (ISSD) with **install all** command is aborted if users with DES privacy protocol are present in the SNMP database. Users need to be reconfigured (using the default AES-128) or deleted. In case of a cold reboot, the SNMP users with DES are deleted.

- When engine ID is configured after configuring the SNMP user, ensure that you perform the following action:
  - After changing the engine ID, reconfigure the SNMP user and the related configuration including group, ACL, along with the password. This avoids authentication failure and impact on the ACL and group attached to the user.

- Beginning with Cisco NX-OS Release 10.3(1)F, SNMP (MIBs – 400G Optic MIB, Switch MIB, Datapath MIBs, Interface MIB) is supported on the Cisco Nexus 9808 platform switches.
- The SVI stats are polled only at an interval of every 120 seconds for SNMP cache.
- Beginning with Cisco NX-OS Release 10.3(3)F, Type-6 encryption for SNMPv3 user password is supported with following limitations:
  - Type-6 encryption is successful only if the following is taken care:
    - **feature password encryption aes {tam}** is enabled.
    - Primary key is configured.
    - The **pwd_type 6** option is specified during SNMPv3 user configuration.
  - Changing the primary key configuration results in SNMP re-encrypting all Type-6 users stored in its database. However, the SNMP functionalities continue to work the same way as before.
  - Primary key configuration is local to the switch. If the user takes the Type-6 configured running data from one switch and applies it on other switch where a different primary key is configured, SNMP features for the same user might not work on the other switch.
  - If Type-6 is configured, ensure to remove the configuration, or reconfigure the Type-6 option before downgrading to the release where Type-6 is not supported.
  - In case of ISSU, if you migrate from an earlier image (where localizedkey, localizedV2key config is present) to a new image where Type-6 encryption is supported, SNMP won't convert the existing keys to Type-6 encryption.
  - Conversion between existing SALT encryption to Type-6 encryption is supported using the **encryption re-encrypt obfuscated** command.
  - ASCII-based reloads through disruptive upgrades and **reload-ascii** commands leads to loss of primary key which would impact the SNMP functionality for the Type-6 users.
  - If a user enforces re-encryption using the **encryption re-encrypt obfuscated** command, then SNMP encrypts all passwords from non-Type-6 SNMP users to Type-6 mode.



Note The SNMP does not support the **encryption delete type6** command.

Default Settings for SNMP

The following table lists the default settings for SNMP parameters.

Parameters	Default
License notifications	Enabled

Configure SNMP

Note Be aware that the Cisco NX-OS commands for this feature may differ from those commands used in Cisco IOS.

Note From Cisco NX-OS release 9.3(7), HMAC-SHA-256 authentication protocol is used for SNMPv3.

The following section includes the various processes involved to configure SNMP.

Configure SNMP Users

You can configure a user for SNMP.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure an SNMP user with authentication and privacy parameters using the command **snmp-server user***name* [**pwd_type 6**] [**auth** { **md5** | **sha** | **sha-224** | **sha-256** | **sha-384** | **sha-512** } *passphrase* [**auto**] [**priv[aes-128]***passphrase*] [**engineID***id*] [**localizedkey**] | [**localizedV2key**]]

Example:

```
switch(config)# snmp-server user Admin pwd_type 6 auth sha abcd1234 priv abcdefgh
```

The passphrase can be any case-sensitive, alphanumeric string up to 64 characters. If you use the **localizedkey** keyword, the passphrase can be any case-sensitive, alphanumeric string up to 130 characters.

sha - Use the HMAC SHA-1 algorithm for authentication.

sha-224 - Use the HMAC SHA-224 algorithm for authentication.

sha-256 - Use the HMAC SHA-256 algorithm for authentication.

sha-384 - Use the HMAC SHA-384 algorithm for authentication.

sha-512 - Use the HMAC SHA-512 algorithm for authentication.

localizedkey - If you use the **localizedkey** keyword, the passphrase can be any case-sensitive, alphanumeric string up to 130 characters. Instead of plain-text password, **hashed** password (copied either from the **show running config** command or generated offline using snmpv3 based open source hash generator tool, see Generate Hashed Password Offline, on page 17) can be configured using the localizedkey keyword.

Note When using a localized key, add `0x` before the hash value, for example, `0x84a716329158a97ac9f22780629bc26c`.

localizedV2key - If the localizedV2key is used, the passphrase can be any case-sensitive, alphanumeric string up to 130 characters, without 0x at the beginning. Collect the localizedv2key using **show run** command, as this is an encrypted data and cannot be generated offline.

The engineID format is a 12-digit, colon-separated decimal number.

Note
- Beginning with Cisco NX-OS Release 10.1(1), AES-128 is the default privacy protocol for SNMPv3.
- Beginning with Cisco NX-OS Release 10.3(3)F, the **pwd_type 6** keyword is supported to provide Type-6 encryption for SNMP users password.

Step 3 Display information about one or more SNMP users using the command **show snmp user**

Example:

```
switch(config)# show snmp user
```

This is an optional step.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Generate Hashed Password Offline

Perform the following steps to generate hashed password offline, using snmpv3-based open source hash generator tool:

Note The IDs mentioned in this procedure are only sample IDs, the purpose of which is only to explain the procedure better.

1. Get the SNMP engineID from the switch.

   switch# **show snmp engineID**

   Sample output:

   ```
   Local SNMP engineID: [Hex] 8000000903D4C93CEA31CC
   [Dec] 128:000:000:009:003:212:201:060:234:049:204
   ```

2. Use an SNMPv3 based open source hash generator to generate offline hashed password.

   Linux$ **snmpv3-hashgen --auth** Hello123 --engine 8000000903D4C93CEA31CC --user1 **--mode priv --hash md5**

   Sample output:

   ```
   User: user1
   Auth: Hello123 / 84a716329158a97ac9f22780629bc26c
   Priv: Hello123 / 84a716329158a97ac9f22780629bc26c
   Engine: 8000000903D4C93CEA31CC
   ESXi USM String: u1/84a716329158a97ac9f22780629bc26c/84a716329158a97ac9f22780629bc26c/priv
   ```

3. Use the auth and priv values to configure the password on the switch.

   snmp-server user user1 **auth md5** 0x84a716329158a97ac9f22780629bc26c **priv des 0x**84a716329158a97ac9f22780629bc26c **localizedkey**

Enforce 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, NX-OS responds with an authorization error for any SNMPv3 PDU request using a security level parameter of either **noAuthNoPriv** or **authNoPriv** .

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Enforce SNMP message encryption for this user using the command **snmp-server user***name***enforcePriv**

Example:

```
switch(config)# snmp-server user Admin enforcePriv
```

Step 3 Enforce SNMP message encryption for all users using the command **snmp-server globalEnforcePriv**

Example:

```
switch(config)# snmp-server globalEnforcePriv
```

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Assign SNMPv3 Users to Multiple Roles

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

Note Only users belonging to a network-admin role can assign roles to other users.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Associate this SNMP user with the configured user role using the command **snmp-server user***namegroup*

Example:

```
switch(config)# snmp-server user Admin superuser
```

Step 3 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Create SNMP Communities

You can create SNMP communities for SNMPv1 or SNMPv2c.

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Create an SNMP community string using the command **snmp-server community***name* { **group***group* | **ro** | **rw** }

Example:

```
switch(config)# snmp-server community public ro
```

Step 3 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Filter SNMP Requests

You can assign an access control list (ACL) to an SNMPv2 community to filter 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)

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Assign an IPv4 ACL to an SNMPv2 community to filter SNMP requests using the command **snmp-server community***name* [**use-ipv4acl***acl-name*]

Example:

```
switch(config)# snmp-server community public use-ipv4acl myacl
```

Step 3 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure SNMP Notification Receivers

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

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure a host receiver for SNMPv1 traps using the command **snmp-server host***ip-address***traps version 1***community* [**udp_port***number*]

Example:

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

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.

Step 3 Configure a host receiver for SNMPv2c traps or informs using the command **snmp-server host***ip-address* { **traps** | **informs** } **version 2c***community* [**udp_port***number*]

Example:

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

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.

Step 4 Configure a host receiver for SNMPv3 traps or informs using the command **snmp-server host***ip-address* { **traps** | **informs** } **version 3** { **auth** | **noauth** | **priv** } *username* [**udp_port***number*]

Example:

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

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 engine ID of the Cisco NX-OS device to authenticate and decrypt the SNMPv3 messages.

Step 5 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Source Interface for SNMP Notifications

You can configure SNMP to use the IP address of an interface as the source IP address for notifications. When a notification is generated, its source IP address is based on the IP address of this configured interface.

You can configure a source interface as follows:

- All notifications sent to all SNMP notification receivers.
- All notifications sent to a specific SNMP notification receiver. This configuration overrides the global source interface configuration.

Note Configuring the source interface IP address for outgoing trap packets does not guarantee that the device will use the same interface to send the trap. The source interface IP address defines the source address inside of the SNMP trap, and the connection is opened with the address of the egress interface as source.

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Send traps messages to this host using the command **snmp-server host***ip-address***source-interface***if-type if-number***traps version 2c***name*

Example:

```
snmp-server host 192.0.2.1 source-interface ethernet 2/1 traps version 2c public
```

This is an optional step. The traps version is the SNMP version to use for notification messages. 2c indicates that SNMPv2c is to be used.

Step 3 Configure SNMP to use the selected VRF to communicate with the host receiver using the command **snmp-server host***ip-address***source-interface***if-type if-number***use-vrf***vrf-name*

Example:

```
snmp-server host 192.0.2.1 source-interface ethernet 2/1 use-vrf default
```

The ip-address can be an IPv4 or IPv6 address. The VRF name can be any alphanumeric string up to 32 characters.



Note This command does not remove the host configuration.

Step 4 Configure a host receiver for SNMPv2c traps or informs using the command **snmp-server host***ip-address***source-interface***if-type if-number* [**udp_port***number*]

Example:

```
switch(config)# snmp-server host 192.0.2.1 source-interface ethernet 2/1
```

The *ip-address* can be an IPv4 or IPv6 address. Use ? to determine the supported interface types. The UDP port number range is from 0 to 65535. This configuration overrides the global source interface configuration.

Step 5 Configure a source interface for sending out SNMPv2c traps or informs using the command **snmp-server source-interface** { **traps** | **informs** } *if-type if-number*

Example:

```
switch(config)# snmp-server source-interface traps ethernet 2/1
```

Use ? to determine the supported interface types.

Step 6 Display information about configured source interfaces using the command **show snmp source-interface**

Example:

```
switch(config)# show snmp source-interface
```

Configure Notification Target User

You must configure a notification target user on the device to send SNMPv3 inform notifications to a notification host receiver.

Cisco NX-OS uses the credentials of the notification target user to encrypt the SNMPv3 inform notification messages to the configured notification host receiver.

Note For authenticating and decrypting the received inform PDU, the notification host receiver should have the same user credentials as configured in Cisco NX-OS to authenticate and decrypt the informs.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure the notification target user with the specified engine ID for the notification host receiver using the command **snmp-server user***name* [**auth** { **md5** | **sha** | **sha-256** } *passphrase* [**auto**] [**priv***passphrase*] [**engineID***id*]

Example:

```
switch(config)# snmp-server user NMS auth sha abcd1234 priv abcdefgh engineID
00:00:00:63:00:01:00:10:20:15:10:03
```

The engine ID format is a 12-digit colon-separated decimal number.

Note Beginning with Release 10.1(1), AES-128 is the default privacy protocol for SNMPv3.

Step 3 Copy the running configuration to the startup configuration using the command**copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure SNMP Notification Receivers with VRFs

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.

You can configure Cisco NX-OS to use a configured VRF to reach the host receiver or to filter notifications based on the VRF in which the notification occurred.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure SNMP to use the selected VRF to communicate with the host receiver using the command [**no**] **snmp-server host***ip-address***use-vrf***vrf-name* [**udp_port***number*]

Example:

```
switch(config)# snmp-server host 192.0.2.1 use-vrf Blue
```

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 the ExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MB.

The **no** form of this command removes the VRF reachability information for the configured host and removes the entry from the ExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MB.

Note This command does not remove the host configuration.

Step 3 Filter notifications to the notification host receiver based on the configured VRF using the command [**no**] **snmp-server host***ip-address***filter-vrf***vrf-name* [**udp_port***number*]

Example:

```
switch(config)# snmp-server host 192.0.2.1 filter-vrf Red
```

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 the ExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MB.

The **no** form of this command removes the VRF filter information for the configured host and removes the entry from the ExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MB.

Note This command does not remove the host configuration.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure SNMP to Send Traps Using an Inband Port

You can configure SNMP to send traps using an inband port. To do so, you must configure the source interface (at the global or host level) and the VRF used to send the traps.

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Globally configure a source interface for sending out SNMP traps using the command **snmp-server source-interface traps***if-type if-number*

Example:

```
switch(config)# snmp-server source-interface traps ethernet 1/2
```

Use ? to determine the supported interface types. You can configure the source interface at the global level or at a host level. When the source interface is configured globally, any new host configuration uses the global configuration to send the traps.

Note To configure a source interface at the host level, use the **snmp-server host***ip-address***source-interface***if-type if-number* command.

Step 3 (Optional) Display information about configured source interfaces using the command **show snmp source-interface**

Example:

```
switch(config)# show snmp source-interface
```

Step 4 Configure SNMP to use the selected VRF to communicate with the host receiver using the command **snmp-server host***ip-address***use-vrf***vrf-name* [**udp_port***number*]

Example:

```
switch(config)# snmp-server host 171.71.48.164 use-vrf default
```

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 the ExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MB.

Note By default, SNMP sends the traps using the management VRF. If you do not want to use the management VRF, you must use this command to specify the desired VRF.

Step 5 Display information about configured SNMP hosts using the command **show snmp host**

Example:

```
switch(config)# show snmp host
```

This is an optional step.

Step 6 Copy the running configuration to the startup configuration using the command**copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Enable SNMP Notifications

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

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

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

Table 3: Enabling SNMP Notifications

MIB	Related Commands
All notifications (except BGP, EIGRP, and OSPF)	**snmp-server enable traps**
CISCO-AAA-SERVER-MIB	**snmp-server enable traps aaa** **snmp-server enable traps aaa server-state-change**
CISCO-BGP4-MIB	**snmp-server enable traps bgp**
CISCO-CALLHOME-MIB	**snmp-server enable traps callhome** **snmp-server enable traps callhome event-notify** **snmp-server enable traps callhome smtp-send-fail**

MIB	Related Commands
CISCO-CONFIG-MAN-MIB	**snmp-server enable traps config** **snmp-server enable traps config ccmCLIRunningConfigChanged**
CISCO-EIGRP-MIB	**snmp-server enable traps eigrp** [*tag*]
CISCO-ERR-DISABLE-MIB	**snmp-server enable traps link cerrDisableInterfaceEventRev1**
ENTITY-MIB, CISCO-ENTITY-SENSOR-MIB	**snmp-server enable traps entity** **snmp-server enable traps entity entity_fan_status_change** **snmp-server enable traps entity entity_mib_change** **snmp-server enable traps entity entity_module_inserted** **snmp-server enable traps entity entity_module_removed** **snmp-server enable traps entity entity_module_status_change** **snmp-server enable traps entity entity_power_out_change** **snmp-server enable traps entity entity_power_status_change** **snmp-server enable traps entity entity_unrecognised_module**
CISCO-FEATURE-CONTROL-MIB	**snmp-server enable traps feature-control** **snmp-server enable traps feature-control FeatureOpStatusChange**
CISCO-HSRP-MIB	**snmp-server enable traps hsrp** **snmp-server enable traps hsrp state-change**
IF-MIB	**snmp-server enable traps link** **snmp-server enable traps link extended-linkDown** **snmp-server enable traps link extended-linkUp** **snmp-server enable traps link cieLinkDown** **snmp-server enable traps link cieLinkUp** **snmp-server enable traps link linkDown** **snmp-server enable traps link linkUp**

MIB	Related Commands
OSPF-MIB, OSPF-TRAP-MIB	**snmp-server enable traps ospf** [*tag*] **snmp-server enable traps ospf lsa** **snmp-server enable traps ospf rate-limit** *rate*
CISCO-RF-MIB	**snmp-server enable traps rf** **snmp-server enable traps rf redundancy_framework**
CISCO-RMON-MIB	**snmp-server enable traps rmon** **snmp-server enable traps rmon fallingAlarm** **snmp-server enable traps rmon hcFallingAlarm** **snmp-server enable traps rmon hcRisingAlarm** **snmp-server enable traps rmon risingAlarm**
SNMPv2-MIB	**snmp-server enable traps snmp** **snmp-server enable traps snmp authentication**
CISCO-MAC-NOTIFICATION-MIB	**snmp-server enable trap link cmn-mac-move-notification**
CISCO-PORT-STORM-CONTROL-MIB	**storm-control action trap**
CISCO-STP-EXTENSIONS-MIB	**snmp-server enable traps stpx stpxMstInconsistencyUpdate**
CISCO-STP-BRIDGE-MIB	**snmp-server enable traps bridge** **snmp-server enable traps bridge newroot** **snmp-server enable traps bridge topologychange**
CISCO-STPX-MIB	**snmp-server enable traps stpx** **snmp-server enable traps stpx inconsistency** **snmp-server enable traps stpx loop-inconsistency** **snmp-server enable traps stpx root-inconsistency**
CISCO-SYSTEM-EXT-MIB	**snmp-server enable traps sysmgr** **snmp-server enable traps sysmgr cseFailSwCoreNotifyExtended**
UPGRADE-MIB	**snmp-server enable traps upgrade** **snmp-server enable traps upgrade UpgradeJobStatusNotify** **snmp-server enable traps upgrade UpgradeOpNotifyOnCompletion**

MIB	Related Commands
VTP-MIB	**snmp-server enable traps vtp** **snmp-server enable traps vtp notifs** **snmp-server enable traps vtp vlancreate** **snmp-server enable traps vtp vlandelete**

Use the following commands in the configuration mode shown to enable the specified notification:

Command	Purpose
snmp-server enable traps **Example:** `switch(config)# snmp-server enable traps`	Enables all SNMP notifications.
snmp-server enable traps aaa [**server-state-change**] **Example:** `switch(config)# snmp-server enable traps aaa`	Enables the AAA SNMP notifications. Optionally, enables the following specific notifications: • **server-state-change**—Enables AAA server state-change notifications.
snmp-server enable traps bgp **Example:** `switch(config)# snmp-server enable traps bgp`	Enables Border Gateway Protocol (BGP) SNMP notifications.
snmp-server enable traps bridge [**newroot**] [**topologychange**] **Example:** `switch(config)# snmp-server enable traps bridge`	Enables STP bridge SNMP notifications. Optionally, enables the following specific notifications: • **newroot**—Enables STP new root bridge notifications. • **topologychange**—Enables STP bridge topology-change notifications.
snmp-server enable traps callhome [**event-notify**] [**smtp-send-fail**] **Example:** `switch(config)# snmp-server enable traps callhome`	Enables Call Home notifications. Optionally, enables the following specific notifications: • **event-notify**—Enables Call Home external event notifications. • **smtp-send-fail**—Enables Simple Mail Transfer Protocol (SMTP) message send fail notifications.
snmp-server enable traps config [**ccmCLIRunningConfigChanged**] **Example:** `switch(config)# snmp-server enable traps config`	Enables SNMP notifications for configuration changes. • **ccmCLIRunningConfigChanged**—Enables SNMP notifications for configuration changes in the running or startup configuration.

Command	Purpose
snmp-server enable traps eigrp [*tag*] **Example:** `switch(config)# snmp-server enable traps eigrp`	Enables CISCO-EIGRP-MIB SNMP notifications.
snmp-server enable traps entity [**entity_fan_status_change**] [**entity_mib_change**] [**entity_module_inserted**] [**entity_module_removed**] [**entity_module_status_change**] [**entity_power_out_change**] [**entity_power_status_change**] [**entity_unrecognised_module**] **Example:** `switch(config)# snmp-server enable traps entity`	Enables ENTITY-MIB SNMP notifications. Optionally, enables the following specific notifications: • **entity_fan_status_change**—Enables entity fan status-change notifications. • **entity_mib_change**—Enables entity MIB change notifications. • **entity_module_inserted**—Enables entity module inserted notifications. • **entity_module_removed**—Enables entity module removed notifications. • **entity_module_status_change**—Enables entity module status-change notifications. • **entity_power_out_change**—Enables entity power-out change notifications. • **entity_power_status_change**—Enables entity power status-change notifications. • **entity_unrecognised_module**—Enables entity unrecognized module notifications.
snmp-server enable traps feature-control [**FeatureOpStatusChange**] **Example:** `switch(config)# snmp-server enable traps feature-control`	Enables feature-control SNMP notifications. Optionally, enables the following specific notifications: • **FeatureOpStatusChange**—Enables feature operation status-change notifications.
snmp-server enable traps hsrp state-change **Example:** `switch(config)# snmp-server enable traps hsrp`	Enables CISCO-HSRP-MIB SNMP notifications. Optionally, enables the following specific notifications: • **state-change**—Enables HSRP state-change notifications.

Command	Purpose
snmp-server enable traps license [**notify-license-expiry**] [**notify-license-expiry-warning**] [**notify-licensefile-missing**] [**notify-no-license-for-feature**] **Example:** `switch(config)# snmp-server enable traps license`	Enables ENTITY-MIB SNMP notifications. Optionally, enables the following specific notifications: • **notify-license-expiry**—Enables license expiry notifications. • **notify-license-expiry-warning**—Enables license expiry warning notifications. • **notify-licensefile-missing**—Enables license file-missing notifications. • **notify-no-license-for-feature**—Enables no-license-installed-for-feature notifications.
snmp-server enable traps link [**cieLinkDown**] [**cieLinkUp**] [**cmn-mac-move-notification**] [**IETF-extended-linkDown**] [**IETF-extended-linkUp**] [**cisco-extended-linkDown**] [**cisco-extended-linkUp**][**linkDown**] [**linkUp**] **Example:** `switch(config)# snmp-server enable traps link`	Enables IF-MIB link notifications. Optionally, enable the following specific notifications: • **IETF-extended-linkDown**—Enables Cisco extended link state down notifications. • **IETF-extended-linkUp**—Enables Cisco extended link state up notifications. • **cmn-mac-move-notification**—Enables MAC address move notifications. • **cisco-extended-linkDown**—Enables Internet Engineering Task Force (IETF) extended link state down notifications. • **cisco-extended-linkUp**—Enables Internet Engineering Task Force (IETF) extended link state up notifications. • **linkDown**—Enables IETF link state down notifications. • **linkUp**—Enables IETF link state up notifications.
snmp-server enable traps ospf [*tag*] [**lsa**] **Example:** `switch(config)# snmp-server enable traps ospf`	Enables Open Shortest Path First (OSPF) notifications. Optionally, enables the following specific notifications: • **lsa**—Enables OSPF link state advertisement (LSA) notifications.

Command	Purpose
snmp-server enable traps rf [**redundancy-framework**] **Example:** `switch(config)# snmp-server enable traps rf`	Enables redundancy framework (RF) SNMP notifications. Optionally, enables the following specific notifications: • **redundancy-framework**—Enables RF supervisor switchover MIB notifications.
snmp-server enable traps rmon [**fallingAlarm**] [**hcFallingAlarm**] [**hcRisingAlarm**] [**risingAlarm**] **Example:** `switch(config)# snmp-server enable traps rmon`	Enables remote monitoring (RMON) SNMP notifications. Optionally, enables the following specific notifications: • **fallingAlarm**—Enables RMON falling alarm notifications. • **hcFallingAlarm**—Enables RMON high-capacity falling alarm notifications. • **hcRisingAlarm**—Enables RMON high-capacity rising alarm notifications. • **risingAlarm**—Enables RMON rising alarm notifications.
snmp-server enable traps snmp [**authentication**] **Example:** `switch(config)# snmp-server enable traps snmp`	Enables general SNMP notifications. Optionally, enables the following specific notifications: • **authentication**—Enables SNMP authentication notifications.
snmp-server enable traps stpx [**inconsistency**] [**loop-inconsistency**] [**root-inconsistency**] **Example:** `switch(config)# snmp-server enable traps stpx`	Enables SNMP STPX notifications. Optionally, enables the following specific notifications: • **inconsistency**—Enables SNMP STPX MIB inconsistency update notifications. • **loop-inconsistency**—Enables SNMP STPX MIB loop-inconsistency update notifications. • **root-inconsistency**—Enables SNMP STPX MIB root-inconsistency update notifications.
snmp-server enable traps syslog [**message-generated**] **Example:** `switch(config)# snmp-server enable traps syslog`	Sends syslog messages as traps to the defined SNMP host. Optionally, enables the following specific notifications: • **message-generated**—Enables software log message generated notifications.

Command	Purpose
snmp-server enable traps sysmgr [**cseFailSwCoreNotifyExtended**] **Example:** `switch(config)# snmp-server enable traps sysmgr`	Enables software change notifications. Optionally, enables the following specific notifications: • **cseFailSwCoreNotifyExtended**—Enables software core notifications.
snmp-server enable traps upgrade [**UpgradeJobStatusNotify**] [**UpgradeOpNotifyOnCompletion**] **Example:** `switch(config)# snmp-server enable traps upgrade`	Enables upgrade notifications. Optionally, enables the following specific notifications: • **UpgradeJobStatusNotify**—Enables upgrade job status notifications. • **UpgradeOpNotifyOnCompletion**—Enables upgrade global status notifications.
snmp-server enable traps vtp [**notifs**] [**vlancreate**] [**vlandelete**] **Example:** `switch(config)# snmp-server enable traps vtp`	Enables VTP notifications. Optionally, enables the following specific notifications: • **notifs**—Enables VTP notifications. • **vlancreate**—Enables VLAN creation notifications. • **vlandelete**—Enables VLAN deletion notifications.
storm-control action traps **Example:** `switch(config-if)# storm-control action traps`	Enables traffic storm control notifications when the traffic storm control limit is reached.

Disable Link Notifications on an Interface

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

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Disable SNMP link-state traps for the interface using the command **interface***typeslot* / *port*

Example:

```
switch(config)# interface ethernet 2/2
```

This command is enabled by default.

Step 3 Disable SNMP link-state traps for the interface using the command **no snmp trap link-status**

Example:

```
switch(config-if)# no snmp trap link-status
```

This command is enabled by default.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config-if)# copy running-config startup-config
```

This is an optional step.

Display SNMP ifIndex for an Interface

The SNMP ifIndex is used across multiple SNMP MIBs to link related interface information.

Procedure

Display the persistent SNMP ifIndex value from the IF-MIB for all interfaces using the command **show interface snmp-ifindex**

Example:

```
switch# show interface snmp-ifindex | grep -i Eth12/1
 Eth12/1 441974784 (0x1a580000)
```

Optionally, use the **|** keyword and the **grep** keyword to search for a particular interface in the output.

Enable One-Time Authentication for SNMP over TCP

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

Procedure

Step 1 Enter the global configuration mode using the following command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Enable a one-time authentication for SNMP over a TCP session using the command **snmp-server tcp-session** [**auth**]

Example:

```
switch(config)# snmp-server tcp-session
```

The default is disabled.

Step 3 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Assign SNMP Device Contact and Location Information

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

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure sysContact, which is the SNMP contact name using the command **snmp-server contact***name*

Example:

```
switch(config)# snmp-server contact Admin
```

Step 3 Configure sysLocation, which is the SNMP location using the command **snmp-server location***name*

Example:

```
switch(config)# snmp-server location Lab-7
```

Step 4 Display the information about one or more destination profiles using the command **show snmp**

Example:

```
switch(config)# show snmp
```

This is an optional step.

Step 5 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure 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.

Before you begin

Determine the logical network entity instance. For more information on VRFs and protocol instances, see the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide or the Cisco Nexus 9000 Series NX-OS Multicast Routing Configuration Guide .

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Map an SNMP context to a protocol instance, VRF, or topology using the command [**no**] **snmp-server context***context-name* [**instance***instance-name*] [**vrf***vrf-name*] [**topology***topology-name*]

Example:

```
switch(config)# snmp-server context public1 vrf red
```

The names can be any alphanumeric string up to 32 characters. The **no** option deletes the mapping between an SNMP context and a protocol instance, VRF, or topology.

Note 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.

Step 3 Map an SNMPv2c community to an SNMP context using the command **snmp-server mib community-map***community-name***context***context-name*

Example:

```
switch(config)# snmp-server mib community-map public context public1
```

This is an optional step. The names can be any alphanumeric string up to 32 characters.

Step 4 Display information about one or more SNMP contexts using the command **show snmp context**

Example:

```
switch(config)# show snmp context
```

This is an optional step.

Step 5 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Disable SNMP

You can disable SNMP on the device.

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Disables SNMP using the command **no snmp-server protocol enable**

Example:

```
switch(config)# no snmp-server protocol enable
```

SNMP is enabled by default.

Note You cannot disable SNMPv1 without disabling SNMPv2. If you want to disable SNMPv1, then configure only SNMPv3, or disable SNMP entirely.

Modify SNMP Server Counter Cache Update Timer

Procedure

Step 1 Enter the global configuration mode using the command **configureterminal**

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Define how long in seconds, the port states are held in the local cache using the command **snmp-servercountercachetimeout***seconds*

Example:

```
switch(config)# snmp-server counter cache timeout 1200
```

The counter cache is enabled by default, and the default cache timeout value is 10 seconds. When disabled, the default cache timeout value is 50 seconds. The range is 1-3600.

Note For end of row (EoR) switching - The range is from 10 to 3600.

Step 3 Display the configured SNMP-server counter cache update timeout value using the command **showrunning-configsnmpall** | **icac**

Example:

```
switch(config)# copy running-config snmp all | i cac
```

Step 4 Disable the counter cache update using the command **nosnmp-servercountercacheenable**

Example:

```
switch(config)# no snmp-server counter cache enable
```

Note When the counter cache update is disabled, the value set in the timeout parameter determines length of time the port states are held the counter cache.

Modify AAA Synchronization Time

You can modify how long Cisco NX-OS holds the synchronized user configuration.

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure the amount of time the AAA synchronized user configuration stays in the local cache using the command **snmp-server aaa-user cache-timeout***seconds*

Example:

```
switch(config)# snmp-server aaa-user cache-timeout 1200
```

The range is from 1 to 86400 seconds. The default is 3600.

Step 3 (Optional) Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

Configure SNMP Local Engine ID

Beginning with Cisco NX-OS Release 7.0(3)I6(1), you can configure the engine ID on a local device.



Note After you configure the SNMP local engine ID, you must reconfigure all SNMP users, any host configured with the V3 users, and the community strings. Beginning with Cisco NX-OS Release 7.0(3)I7(1), you need to reconfigure only the SNMP users and community strings.

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Change the SNMP engine ID of the local device using the command **snmp-server engineID local***engineid-string*

Example:

```
switch(config)# snmp-server engineID local AA:BB:CC:1A:2C:10
```

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, 80:00:02:b8:04:61:62:63.

Step 3 Display the identification of the configured SNMP engine using the command **show snmp engineID**

Example:

```
switch(config)# show snmp engineID
```

Step 4 Disable the local engine ID and the default auto-generated engine ID is configured using the command [**no**] **snmp-server engineID local***engineid-string*

Example:

```
switch(config)# no snmp-server engineID local AA:BB:CC:1A:2C:10
```

Step 5 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

Verify SNMP Configuration

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

Command	Purpose
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. SNMP users brought into 10.1(1), from releases prior to 10.1(1), are displayed with the configured privacy protocol, AES-128 or DES. New users (Release 10.1(1) and later) are by default configured with AES-128 protocol. Beginning with 9.3(8) release, SNMPv3 users under show run will be represented in SALT format instead of hash.
show snmp	Displays the SNMP status.
show snmp community	Displays the SNMP community strings. **Note** If the name of the SNMP context in the **snmp-server mib community-map** command is more than 11 characters, the output of the **show snmp community** command is displayed in a vertical format instead of a tabular format.

Command	Purpose
show snmp context	Displays the SNMP context mapping.
show snmp engineID	Displays the SNMP engineID.
show snmp group	Displays SNMP roles.
show snmp host	Displays information about configured SNMP hosts.
show snmp session	Displays SNMP sessions.
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.

Configuration Examples

Configure Cisco NX-OS to send the Cisco linkUp or Down notifications to one notification host receiver using the Blue VRF and defines two SNMP users, Admin and NMS

```
configure terminal
snmp-server contact Admin@company.com
snmp-server user Admin auth sha abcd1234 priv abcdefgh
snmp-server user NMS auth sha abcd1234 priv abcdefgh engineID
00:00:00:63:00:01:00:22:32:15:10:03
snmp-server host 192.0.2.1 informs version 3 auth NMS
snmp-server host 192.0.2.1 use-vrf Blue
snmp-server enable traps link cisco
```

Configure SNMP to send traps using an inband port configured at the host level

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# snmp-server host 171.71.48.164 version 2c public
switch(config)# snmp-server host 171.71.48.164 source-interface ethernet 1/2
switch(config)# show snmp host
-------------------------------------------------------------------
Host Port Version Level Type SecName
-------------------------------------------------------------------
171.71.48.164 162 v2c noauth trap public
Source interface: Ethernet 1/2
-------------------------------------------------------------------
switch(config)# snmp-server host 171.71.48.164 use-vrf default
switch(config)# show snmp host
-------------------------------------------------------------------
Host Port Version Level Type SecName
-------------------------------------------------------------------
171.71.48.164 162 v2c noauth trap public
Use VRF: default
Source interface: Ethernet 1/2
```

```
-----------------------------------------------------------------
```

Configure SNMP to send traps using a globally configured inband port

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# snmp-server source-interface traps ethernet 1/2
switch(config)# show snmp source-interface
-----------------------------------------------------------------
Notification source-interface
-----------------------------------------------------------------
trap Ethernet1/2
inform -
-----------------------------------------------------------------
switch(config)# snmp-server host 171.71.48.164 use_vrf default
switch(config)# show snmp host
-----------------------------------------------------------------
Host Port Version Level Type SecName
-----------------------------------------------------------------
171.71.48.164 162 v2c noauth trap public
Use VRF: default
Source interface: Ethernet 1/2
-----------------------------------------------------------------
```

Map VRF red to the SNMPv2c public community string

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# vrf context red
switch(config-vrf)# exit
switch(config)# snmp-server context public1 vrf red
switch(config)# snmp-server mib community-map public context public1
```

Map OSPF instance Enterprise to the same SNMPv2c public community string

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# feature ospf
switch(config)# router ospf Enterprise
switch(config-router)# exit
switch(config)# snmp-server context public1 instance Enterprise
switch(config)# snmp-server mib community-map public context public1
```

Additional References

Related Documents

Related Topic	Document Title
IP ACLs and AAA	*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*
MIBs	*Cisco Nexus 7000 Series and 9000 Series NX-OS MIB Quick Reference*

Related Topic	Document Title
RFC 3414	*User-based Security Model (USM) for Version 3 of the Simple Network Management Protocol (SNMPv3)*
RFC 3415	*View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*
MIBs related to SNMP	To locate and download supported MIBs, go to the following URL: https://cisco.github.io/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html

RFCs

RFC	Title
RFC 3414	*User-based Security Model (USM) for Version 3 of the Simple Network Management Protocol (SNMPv3)*
RFC 3415	*View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*

MIBs

MIBs	MIBs Link
MIBs related to SNMP	To locate and download supported MIBs, go to the following https://cisco.github.io/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html