Simple Network Management Protocol Configuration Guide, Cisco Catalyst IE9300 Rugged Series Switches

Cisco Catalyst IE9300 Rugged Series
Updated August 17, 2026

© 2025-2026 Cisco Systems, Inc. All rights reserved.

Full Cisco Trademarks with Software License

Full Cisco Trademarks with Software License

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

1 Configuring Simple Network Management Protocol

Topics:

- SNMP versions and security models
- Guidelines and limitations
- SNMP overview
- Configure SNMP
- SNMP examples
- Monitor SNMP status

Configuring Simple Network Management Protocol (SNMP)

SNMP versions and security models

This topic explains the prerequisites for SNMP, including supported SNMP versions, their features, and security models and levels.

This software release supports SNMPv1, SNMPv2C, and SNMPv3. Each version offers distinct features and security models to manage network devices effectively.

Supported SNMP versions

- **SNMPv1:** The Simple Network Management Protocol, a Full Internet Standard, defined in RFC 1157.
- **SNMPv2C:** The community-string-based administrative framework for SNMPv2, an experimental internet protocol defined in RFC 1901. It retains the bulk retrieval and improved error handling of SNMPv2Classic.
- **SNMPv3:** An interoperable standards-based protocol defined in RFCs 2273 to 2275. It provides secure access to devices through these features.
  - Message integrity ensures that a packet was not tampered with in transit.
  - Authentication determines that the message is from a valid source.
  - Encryption prevents unauthorized sources from reading packet contents.

> **Note**
>
> Both SNMPv1 and SNMPv2C use a community-based form of security. The management station access is defined by an IP address access control list and a password.

SNMPv3 security models and levels

SNMPv3 provides security models and levels. A security model is an authentication strategy set up for a user and their group. A security level defines the type of security permitted in a security model. Available security models include SNMPv1, SNMPv2C, and SNMPv3.

> **Note**
>
> To select encryption, enter the **priv** keyword.

The table identifies characteristics and compares combinations of security models and levels:

Table 1: SNMP security models and levels

Model	Level	Authentication	Encryption	Result
SNMPv1	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
SNMPv2C	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
SNMPv3	noAuthNoPriv	Username	No	Uses a username match for authentication.

Model	Level	Authentication	Encryption	Result
SNMPv3	authNoPriv	MD5 or SHA	No	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms.
SNMPv3	authPriv	MD5 or SHA	Data Encryption Standard (DES) or Advanced Encryption Standard (AES)	Provides authentication based on HMAC-MD5 or HMAC-SHA algorithms and allows this encryption: • DES 56-bit encryption. • 3DES 168-bit encryption. • AES 128-bit, 192-bit, or 256-bit encryption.

Guidelines and limitations

This topic explains restrictions for SNMP, including version-specific limitations and device-specific support details.

- SNMPv1 does not support informs.
- To prevent SNMPv3 authentication failures, manually configure the SNMP engineID before adding SNMPv3 users. This process ensures that the user is associated with the correct engineID, enabling consistent device management.

SNMP overview

Explains what SNMP protocols are and their role in network device management.

SNMP is a network management protocol used to monitor and manage network devices.

SNMP system components

This topic explains the Simple Network Management Protocol (SNMP), its components, and how it facilitates communication between managers and agents for network management and monitoring.

SNMP system consists of these components:

- **SNMP manager:** A system, often part of a network management system (NMS), that requests or changes values in the MIB.
- **SNMP agent:** A software component residing on the device that gathers data from the MIB and responds to manager requests.
- **Management information base (MIB):** A repository that stores information about device parameters and network data.The agent and MIB reside on the device.

SNMP communication

SNMP enables managers to request or change MIB variable values on agents, and agents to respond or send traps to managers.

- The SNMP agent contains MIB variables whose values the SNMP manager can request or change.
- The agent gathers data from the MIB, which stores information about device parameters and network data.
- An agent can send unsolicited traps to the manager, alerting the SNMP manager to network conditions.

Traps can indicate events such as improper user authentication, device restarts, link status changes, MAC address tracking, TCP connection closure, or loss of connection to a neighbor.

SNMP manager functions

This topic explains the functions of an SNMP manager, including the operations it performs using information in the MIB and important recommendations for optimal performance.

- The SNMP manager uses information in the MIB to perform various operations for network management.
- Key SNMP operations include retrieving, storing, and responding to variable values, as well as handling unsolicited messages from agents.

SNMP manager operations

The table describes the main SNMP operations performed by the SNMP manager using the MIB.

Table 2: SNMP Operations

Operation	Description
get-request	Retrieves a value from a specific variable.
get-next-request	Retrieves a value from a variable within a table. With this operation, an SNMP manager does not need to know the exact variable name. A sequential search is performed to find the needed variable from within a table.
get-bulk-request	Retrieves large blocks of data, such as multiple rows in a table, to minimize transmissions. This operation requires SNMPv2 or later.
get-response	Replies to a get-request, get-next-request, or set-request sent by an NMS.
set-request	Stores a value in a specific variable.
trap	Sends an unsolicited message from an SNMP agent to an SNMP manager when an event occurs.

Note

Exclude the **ciscoFlashFileDate** MIB object from SNMP manager queries to prevent performance issues. Although this object is published in the MIB, it is not supported on the product.

SNMP agent functions

This topic explains the functions of the SNMP agent, including how it processes requests from SNMP managers, tracks polling information, and sends trap notifications for significant events.

The SNMP agent is a software component that receives and responds to requests from one or more SNMP managers. The agent tracks the NMS IP address, the number of polls, and the polling timestamp for both IPv4 and IPv6 servers.

SNMP agent request handling

The SNMP agent performs these functions in response to NMS requests:

- **Get a MIB variable:** The agent retrieves the value of the requested MIB variable and sends the value to the NMS.
- **Set a MIB variable:** The agent retrieves the value of the requested MIB variable and sends the value to the NMS.

SNMP statistics commands

Use these commands to manage SNMP statistics:

- The **show snmp stats hosts** command to display the list of SNMP manager requests in the queue.
- The **clear snmp stats hosts** command to clear the queue.

SNMP trap notifications

The SNMP agent sends unsolicited trap messages to notify an NMS that a significant event has occurred. Trap conditions include:

- Port or module status changes (up or down)
- Spanning-tree topology changes
- Authentication failures

SNMP community strings

This topic explains SNMP community strings, which authenticate access to MIB objects and act as embedded passwords. It describes their attributes and how they are used for device management and security.

SNMP community strings authenticate access to MIB objects and serve as embedded passwords. For an NMS to access a device, the community string configured on the NMS must match one of the community strings defined on the device.

SNMP community string attributes

SNMP community strings have these attributes. Their configuration affects how management stations interact with devices.

- **Read-only (RO):** Grants authorized management stations read access to all MIB objects but denies write access.
- **Read-write (RW):** Grants authorized management stations read and write access to all MIB objects but denies access to the community strings.

Note

When you create a cluster, the command device manages message exchange between member devices and the SNMP application. Network Assistant appends the member device number (@esN, where N is the device number) to the first configured RW and RO community strings on the command device. It then copies these strings to the member device.

SNMP MIB variable access

This topic explains how SNMP MIB variables are accessed and used by NMS software to monitor, configure, and troubleshoot network devices through SNMP agents and managers.

SNMP MIB Variables Access refers to the process by which NMS software interacts with device MIB variables to monitor and manage network devices.

- NMS software uses MIB variables to set device parameters and poll devices for information.
- SNMP agents gather data from the MIB and send traps (notifications) to the SNMP manager about network events.
- SNMP agents respond to MIB-related queries from the SNMP manager in **get-request** , **get-next-request** , and **set-request** formats.

Accessing SNMP MIB variables

SNMP MIB variable access involves communication between NMS software, SNMP agents, and SNMP managers to monitor and manage network devices.

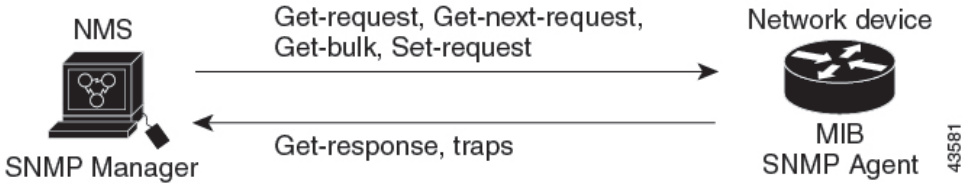
- NMS software polls devices for specific information using MIB variables.
- Results from polls can be displayed as graphs and analyzed for troubleshooting, performance monitoring, and configuration verification.
- SNMP agents send traps to notify the SNMP manager of network events such as authentication failures, restarts, link status changes, and MAC address tracking.

SNMP queries

SNMP agents respond to queries from the SNMP manager using specific request formats.

1. **get-request**
2. **get-next-request**
3. **set-request**

Figure 1: SNMP Network



SNMP Flash MIB

This topic explains the SNMP Flash MIB, which is used to query flash file data on Cisco devices and describes recent changes to its file listing limitations and usage requirements.

The Flash MIB queries flash file data from Cisco devices. The Flash MIB fetches all files from the flash file system, removing the previous 100-file limitation per partition.

Use the **snmp mib flash cache** command to prefetch all files into the local Flash MIB cache before performing a Flash MIB walk. Retrieving all files from the file system increases the time required to complete a Flash MIB walk.

Flash MIB usage and recommendations

Follow these guidelines to maintain system performance and prevent SNMP walk timeouts:

- Use the **snmp mib flash cache** command with caution, as it may impact CPU performance.
- Set the SNMP walk timeout period to at least 10 seconds and the default retry interval to 5 seconds. These values help prevent SNMP walks from timing out.

SNMP notifications

This topic explains SNMP notifications, including how devices send notifications as traps or inform requests, and the trade-offs between reliability and resource usage for each method.

SNMP notifications are messages that a device sends to SNMP managers when specific events occur. Devices can send these messages as traps or inform requests.

- Traps are notifications sent without acknowledgment. They may not reach their destination.
- Inform requests require acknowledgment and can be resent if not received, making them more reliable.
- Choosing between traps and informs involves a trade-off between reliability and resource consumption.

Notification reliability and resource trade-offs

Traps and informs require a trade-off between reliability and resource consumption.

- **Traps:** Unreliable because the receiver does not acknowledge receipt. The sender discards the trap immediately after sending it.
- **Informs:** Reliable because the SNMP manager acknowledges receipt with an SNMP response protocol data unit (PDU). The sender holds the inform request in memory until it receives a response or the request times out. If the sender does not receive a response, it resends the inform request.

> **Note**
>
> SNMPv1 does not support informs.

Characteristic	Traps	Informs
Reliability	Low	High
Acknowledgment	No	Yes
Resource consumption	Low	High
Network traffic	Low	High (due to retries)

Configuration principle

Follow these guidelines to choose the appropriate notification type:

- Use inform requests if it is critical that the SNMP manager receives every notification.
- Use traps if network traffic or device memory is a concern and guaranteed delivery is not required.

SNMP ifIndex MIB object values

This topic explains how the SNMP agent's IF-MIB module assigns ifIndex numbers to physical interfaces during initialization, and how these values may change after a reboot depending on driver initialization order.

The SNMP ifIndex MIB object value is a unique identifier assigned by the IF-MIB module to each physical interface as drivers are initialized after a device reboot.

- The IF-MIB module assigns ifIndex numbers on a first-come-first-served basis as interface drivers register.

- Driver initialization order can vary between reboots, causing the same physical interface to receive a different ifIndex number.
- IfIndex persistency must be enabled to maintain consistent ifIndex assignments across reboots.

SNMP ENTITY-MIB identifiers

This topic explains how the *entPhysicalIndex* number in ENTITY-MIB changes after an OIR operation, even if the same entity is reinserted. It describes the assignment and management of unique index numbers for physical entities.

ENTITY-MIB contains information for managing physical entities such as field-replaceable units (FRUs) on a device. Each entity is identified by a unique index number, *entPhysicalIndex*, which is used to access information about the entity in current and other MIBs.

- An online insertion and removal (OIR) operation results in the entity being assigned the next available *entPhysicalIndex* number.
- The *entPhysicalIndex* changes after OIR, even if the same entity is reinserted.

SNMP and Syslog over IPv6

This topic explains how SNMP and syslog over IPv6 support both IPv4 and IPv6 transports, enabling network management and logging capabilities for dual-stack environments.

SNMP and syslog over IPv6 are network management and logging features that support both IPv4 and IPv6 transports. These features:

SNMP and syslog over IPv6 provide these features:

- Support for both IPv4 and IPv6
- IPv6 transport for SNMP and modification of the SNMP agent to support traps for an IPv6 host
- SNMP- and syslog-related MIBs to support IPv6 addressing
- Configuration of IPv6 hosts as trap receivers

Features and operations

SNMP actions that support IPv6 transport management include:

1. Opens User Datagram Protocol (UDP) SNMP socket with default settings
2. Provides a new transport mechanism called *SR_IPV6_TRANSPORT*
3. Sends SNMP notifications over IPv6 transport
4. Supports SNMP-named access lists for IPv6 transport
5. Supports SNMP proxy forwarding using IPv6 transport
6. Verifies SNMP Manager feature works with IPv6 transport

Note

For information on SNMP over IPv6, including configuration procedures, see the "Managing Cisco IOS Applications over IPv6" chapter in the *Cisco IOS IPv6 Configuration Library* on the Cisco website. For information about syslog over IPv6, including configuration procedures, see the "Implementing IPv6 Addressing and Basic Connectivity" chapter in the *Cisco IOS IPv6 Configuration Library* on the Cisco website.

Default SNMP configuration

This topic explains the default SNMP configuration settings, including the default state of the SNMP agent, trap receiver, traps, version, authentication, and notification type on the device.

If the device starts and no snmp-server global configuration commands are present in the startup configuration, the device operates using the default SNMP settings.

Table 3: Default SNMP Configuration Table

Feature	Default Setting
SNMP agent	Disabled. This is the default when the device starts and the startup configuration does not have any **snmp-server** global configuration commands.
SNMP trap receiver	None configured.
SNMP traps	None enabled except the trap for TCP connections (tty).
SNMP version	If no version keyword is present, the default is Version 1.
SNMPv3 authentication	If no keyword is entered, the default is the **noauth** (noAuthNoPriv) security level.
SNMP notification type	If no type is specified, all notifications are sent.

SNMP configuration guidelines

This topic explains the guidelines for configuring SNMP on a device, including required commands, group and user relationships, and important considerations for SNMP engine IDs and UDP port configuration.

These SNMP configuration guidelines define the requirements and best practices for enabling and managing SNMP agents on a device.

- At least one global SNMP command (**snmp-server host** , **snmp-server user** , **snmp-server community** , or **snmp-server manager**) must be configured to open SNMP UDP ports 161 and 162.
- An SNMP *group* maps users to views. A *user* is a group member, a *host* receives traps, and an *engine ID* identifies the SNMP engine.
- Proper configuration of SNMP groups, users, engine IDs, and UDP ports is essential for secure and functional SNMP operation.

When configuring SNMP, follow these guidelines to ensure correct and secure operation:

- When configuring an SNMP group, do not specify a notify view. The **snmp-server host** command auto-generates a notify view for the user and adds it to the group. Modifying the group's notify view affects all users in that group.
- To configure a remote user, specify the IP address or port number for the remote SNMP agent where the user resides.
- Before configuring remote users for an agent, configure the SNMP engine ID using the **snmp-server engineID** command with the **remote** option. The remote agent's SNMP engine ID and user password are used to compute authentication and privacy digests. If the remote engine ID is not configured first, the command fails.
- When configuring SNMP informs, configure the SNMP engine ID for the remote agent in the SNMP database before sending proxy requests or informs to it.
- If a local user is not associated with a remote host, the device does not send informs for the **auth** (authNoPriv) and **priv** (authPriv) authentication levels.
- Changing the SNMP engine ID value has significant effects. A user's password is converted to an MD5 or SHA digest based on the password and local engine ID, and the command-line password is destroyed as required by RFC 2274. If the engine ID changes, SNMPv3 user digests become invalid and users must be reconfigured using the **snmp-server user** *username* command. Community strings must also be reconfigured when the engine ID changes.
- When you configure the SNMP server host with the default UDP port 162, the output of the **show running-config** command does not display the UDP port value. If you specify a UDP port other than the default using the **snmp-server host** { *host-addr* } *community-string* **udp-port** *value* command, the UDP port number will be displayed in the **show running-config** output. You can configure the **snmp-server host** command with or without the default UDP port 162, but not both simultaneously.

Correct configuration examples:

- Device(config)# **snmp-server host 10.10.10.10 community udp-port 163**
- Device(config)# **snmp-server host 10.10.10.10 community**
- Device(config)# **snmp-server host 10.10.10.10 community udp-port 163**
- Device(config)# **snmp-server host 10.10.10.10 community udp-port 162**

Incorrect configuration examples:

- Device(config)# **snmp-server host 10.10.10.10 community udp-port 163**
- Device(config)# **snmp-server host 10.10.10.10 community**
- Device(config)# **snmp-server host 10.10.10.10 community udp-port 162**
- Device(config)# **snmp-server host 10.10.10.10 community udp-port 163**
- Device(config)# **snmp-server host 10.10.10.10 community udp-port 162**
- Device(config)# **snmp-server host 10.10.10.10 community**

Configure SNMP

This topic explains how to configure SNMP to enable standardized network device monitoring and management by setting community strings, user groups, UDP ports, notifications, and system identification details.

Configure SNMP to enable standardized network device monitoring and management by setting community strings, user groups, UDP ports, notifications, and system identification details.

SNMP community strings

This topic explains SNMP community strings, which authenticate access to MIB objects and act as embedded passwords. It describes their attributes and how they are used for device management and security.

SNMP community strings authenticate access to MIB objects and serve as embedded passwords. For an NMS to access a device, the community string configured on the NMS must match one of the community strings defined on the device.

SNMP community string attributes

SNMP community strings have these attributes. Their configuration affects how management stations interact with devices.

- **Read-only (RO):** Grants authorized management stations read access to all MIB objects but denies write access.
- **Read-write (RW):** Grants authorized management stations read and write access to all MIB objects but denies access to the community strings.

Note

When you create a cluster, the command device manages message exchange between member devices and the SNMP application. Network Assistant appends the member device number (@esN, where N is the device number) to the first configured RW and RO community strings on the command device. It then copies these strings to the member device.

Configure SNMP groups and users

You can configure SNMP groups and users on a device. This includes mapping SNMP users to SNMP views and adding new users to SNMP groups.

You can specify an identification name (engine ID) for the local or remote SNMP server engine on the device. You can configure an SNMP server group to map SNMP users to SNMP views. You can also add new users to the SNMP group.

Complete these steps to configure SNMP groups and users on the device.

Procedure

1. Use the **enable** command to enter privileged EXEC mode.

   Example

   ```
   Device# enable
   ```

   Enter your password if prompted.

2. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

   Enters global configuration mode.

3. Use **snmp-server engineID local** *engineid-string* **remote** *ip-address* **udp-port** *port-number engineid-string* command to configure a name for the local SNMP engine.

Example

```
Device(config)# snmp-server engineID local 1234
```

The device configures the engine ID as "123400000000000000000000" (trailing zeros can be omitted). You do not need to specify the entire 24-character engine ID value if it has trailing zeros. You only need to specify the part of the engine ID before the trailing zeros.

```
Device(config)# snmp-server group public v2c access lmnop
```

Configures the engine ID for a remote SNMP server at IP 10.1.1.1 on UDP port 162. If you select **remote**, specify the *ip-address* of the device that contains the remote copy of SNMP and the optional UDP port on the remote device. The default value is 162.

4. Use the **snmp-server group** *group-name* **v1 v2c v3 auth noauth priv** **read** *readview* **write** *writeview* **notify** *notifyview* **access** *access-list*

Example

```
Device(config)# snmp-server group public v2c access lmnop
```

Note

Creates an SNMP group named *public* using SNMP version 2c with access list *lmnop*. Security models include:

- **v1**: least secure
- **v2c**: supports informs and wider integers
- **v3**: most secure, requires auth level
  - **auth** –Enables the Message Digest 5 (MD5) and the Secure Hash Algorithm (SHA) packet authentication.
  - **noauth** –Enables the noAuthNoPriv security level. This is the default if no keyword is specified.
  - **priv** –Enables Data Encryption Standard (DES) packet encryption (also called privacy).

Optional parameters (not to exceed 64 characters):

- **read** *readview* : view for read-only access
- **write** *writeview* : view for write access
- **notify** *notifyview* : view for notifications
- **access** *access-list* : access control list name

5. Use the **snmp-server user** *username group-name* **remote** *host* **udp-port** *port* **v1 access** *access-list* **v2c access** *access-list* **v3 encrypted access** *access-list* **auth md5 sha** *auth-password* *priv* **des 3des aes 128 192 256** *priv-password* command to add a new user to an SNMP group with version and authentication options.

Example

```
Device(config)# snmp-server user Pat public v2c
```

Adds user *Pat* to group *public* using SNMP version 2c. For SNMPv3, additional options include:

- **encrypted**: password is encrypted
- **auth md5 sha** *auth-password*: authentication protocol and password
- *priv* **des 3des aes 128 192 256** *priv-password*: privacy encryption and password
- **remote** *host* **udp-port** *port* : specify remote SNMP entity and optional UDP port
- **access** *access-list*: specify access list

> **Note**
>
> Algorithms md5, des, and 3des require compliance shield enabled using `crypto engine compliance shield enable` command and device reboot.

6. Use **end** command to exit global configuration mode.

   Example

   ```
   Device(config)# end
   ```

   Returns to privileged EXEC mode.

7. Use the **show running-config** command to verify your SNMP configuration.

   Example

   ```
   Device# show running-config
   ```

8. (Optional) Use **copy running-config startup-config** command to save the configuration.

   Example

   ```
   Device# copy running-config startup-config
   ```

   Saves the running configuration to startup configuration.

Open or close SNMP UDP ports

Learn how to open or close SNMP UDP ports 161 and 162, which are used for polling and sending notifications from the agent to the server, to enhance device security.

The SNMP process uses UDP ports 161 and 162, where port 161 is designated for polling the device, and port 162 is used for sending notifications from the agent to the server. These SNMP UDP ports remain closed by default and open only when one of the required SNMP configuration commands is applied. This approach enhances security by ensuring the device listens on these ports only when necessary, reducing exposure to unnecessary network traffic.

To open SNMP UDP ports, start in user EXEC mode and execute these steps:

Procedure

1. Use the **enable** command to enter privileged EXEC mode.

   Example

   ```
   Device# enable
   ```

   Enter your password if prompted.

2. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

3. Use the **snmp-server** { **host** | **user** | **community** | **manager** } commands to open ports 161 and 162.

   Example

   ```
   Device(config)# snmp-server host 10.10.10.10
   ```

   Configuring any one of these options opens both UDP ports. To close the ports, enter the no form of all configured options. The ports remain open if at least one of these commands is active. Using the **no snmp-server** command without options shuts down the entire SNMP process.

4. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Device(config)# end
   ```

5. Use the **show udp** command to verify that ports 161 and 162 are listening.

   Example

   ```
   Device# show udp
   ```

   If the requisite commands are configured, ports 161 and 162 will show a **listen** status under the remote field.

6. (Optional) Use the **copy running-config startup-config** command to save the configuration.

   Example

   ```
   Device# copy running-config startup-config
   ```

This procedure opens SNMP UDP ports securely and only when required. It aligns with best practices for device security and management.

SNMP notifications

This topic explains SNMP notifications, including how devices send notifications as traps or inform requests, and the trade-offs between reliability and resource usage for each method.

SNMP notifications are messages that a device sends to SNMP managers when specific events occur. Devices can send these messages as traps or inform requests.

- Traps are notifications sent without acknowledgment. They may not reach their destination.
- Inform requests require acknowledgment and can be resent if not received, making them more reliable.
- Choosing between traps and informs involves a trade-off between reliability and resource consumption.

Notification reliability and resource trade-offs

Traps and informs require a trade-off between reliability and resource consumption.

- **Traps:** Unreliable because the receiver does not acknowledge receipt. The sender discards the trap immediately after sending it.
- **Informs:** Reliable because the SNMP manager acknowledges receipt with an SNMP response protocol data unit (PDU). The sender holds the inform request in memory until it receives a response or the request times out. If the sender does not receive a response, it resends the inform request.

> **Note**
>
> SNMPv1 does not support informs.

Characteristic	Traps	Informs
Reliability	Low	High
Acknowledgment	No	Yes
Resource consumption	Low	High
Network traffic	Low	High (due to retries)

Configuration principle

Follow these guidelines to choose the appropriate notification type:

- Use inform requests if it is critical that the SNMP manager receives every notification.
- Use traps if network traffic or device memory is a concern and guaranteed delivery is not required.

Set the SNMP agent contact and location information

Learn how to set the system contact and location of the SNMP agent so that these descriptions can be accessed through the configuration file.

Set the system contact and location information for the SNMP agent to ensure these descriptions are accessible through the configuration file.

Procedure

1. Use the **enable** command to enter privileged EXEC mode.

   Example

   ```
   Device# enable
   ```

   Enter your password if prompted.

2. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

3. Use the **snmp-server contact** *text* command to set the system contact string.

   Example

   ```
   Device(config)# snmp-server contact Dial System Operator at beeper 21555
   ```

4. Use the **snmp-server location** *text* command to set the system location string.

   Example

   ```
   Device(config)# snmp-server location Building 3/Room 222
   ```

   Sets the system location string.

5. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Device(config)# end
   ```

6. Use the **show running-config** command to verify your entries.

   Example

   ```
   Device# show running-config
   ```

7. (Optional) Use the **copy running-config startup-config** command to save your entries in the configuration file.

   Example

   ```
   Device# copy running-config startup-config
   ```

Limit TFTP servers for SNMP configuration transfers

Learn how to limit the TFTP servers used for saving and loading configuration files through SNMP to the servers specified in an access list.

Follow these steps to limit the TFTP servers used for saving and loading configuration files through SNMP to the servers specified in an access list.

Procedure

1. Use the **enable** command to enter privileged EXEC mode.

   Example

   ```
   Device# enable
   ```

   Enter your password if prompted.

2. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

   Enters global configuration mode.

3. Use the **snmp-server tftp-server-list** *access-list-number* command to limit the TFTP servers used for saving and loading configuration files through SNMP to those specified in the access list.

   Example

   ```
   Device(config)# snmp-server tftp-server-list 44
   ```

   For *access-list-number*, specify an IP standard access list numbered from 1 to 99 or 1300 to 1999.

4. Use the **access-list** *access-list-number* **deny permit** *source source-wildcard* command to create a standard access list that defines which TFTP servers are permitted or denied access. Repeat this command as needed for multiple entries.

   Example

   ```
   Device(config)# access-list 44 permit 10.1.1.2
   ```

   - *access-list-number* must match the value used in step 3.
   - **deny** denies access if conditions match; **permit** keyword allows access.
   - *source* is the IP address of the TFTP server allowed or denied.
   - *source* is optional and specifies wildcard bits in dotted decimal notation to ignore bits in the source IP address.

> **Note**
>
> The access list always ends with an implicit deny for all other addresses.

5. Use **end** command to exit global configuration mode.

   Example

   ```
   Device(config)# end
   ```

   Returns to privileged EXEC mode.

6. Use the **show running-config** command to verify your SNMP configuration.

   Example

   ```
   Device# show running-config
   ```

7. (Optional) Use **copy running-config startup-config** command to save the configuration.

   Example

   ```
   Device# copy running-config startup-config
   ```

   Saves the running configuration to startup configuration.

Disable the SNMP agent

Learn how to disable the SNMP agent on your device using the **no snmp-server** command. This procedure applies to all SNMP agent versions and ensures the SNMP process is shut down.

Use this procedure to disable the SNMP agent on your device.

This procedure applies to all SNMP versions (Version 1, Version 2c, and Version 3) and ensures that the SNMP process on the device is completely shut down.

You can re-enable all versions of the SNMP agent by using any one of these commands in global configuration mode.

- **snmp-server host**
- **snmp-server user**
- **snmp-server community**
- **snmp-server manager**

Note

There is no Cisco IOS command specifically designated for enabling SNMP.

Before you begin

The SNMP agent must be enabled before it can be disabled. The SNMP agent is enabled by the first **snmp-server** global configuration command entered on the device.

Procedure

1. Use the **enable** command to enter privileged EXEC mode.

   Example

   ```
   Device# enable
   ```

   Enter your password if prompted.

2. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

   Enters global configuration mode.

3. Use the **no snmp-server** command to disable the SNMP agent operation.

   Example

   ```
   Device(config)# no snmp-server
   ```

   Disables the SNMP agent operation.

4. Use **end** command to exit global configuration mode.

   Example

   ```
   Device(config)# end
   ```

   Returns to privileged EXEC mode.

5. Use the **show running-config** command to verify your SNMP configuration.

   Example

   ```
   Device# show running-config
   ```

6. (Optional) Use **copy running-config startup-config** command to save the configuration.

Example

```
Device# copy running-config startup-config
```

Saves the running configuration to startup configuration.

The SNMP agent is disabled and the SNMP process is shut down. All SNMP agent versions are no longer operational on the device.

SNMP examples

This topic explains how to configure SNMP on a device, including enabling SNMP, setting community strings, sending traps, and associating users with remote hosts using various SNMP versions and security options.

SNMP configuration involves enabling SNMP, setting community strings, sending traps to hosts, and managing user access and authentication.

Here are a few examples that demonstrate how to configure SNMP on Cisco devices for various use cases:

- **Enable all SNMP versions with read-only access**

  Any SNMP manager can access all objects with read-only permissions using the community string **public**. This configuration does not send any traps from the device.

  ```
  Device(config)# snmp-server community public
  ```

- **Enable SNMP with VTP traps to specific hosts**

  Any SNMP manager can access all objects with read-only permission using the community string **public**. The device sends VTP traps to hosts `192.180.1.111` and `192.180.1.33` using SNMPv1, and to host `192.180.1.27` using SNMPv2C. The community string **public** is sent with the traps.

  ```
  Device(config)# snmp-server community public
  Device(config)# snmp-server enable traps vtp
  Device(config)# snmp-server host 192.180.1.27 version 2c public
  Device(config)# snmp-server host 192.180.1.111 version 1 public
  Device(config)# snmp-server host 192.180.1.33 public
  ```

- **Restrict SNMP access using an access list**

  Members of access list 4 who use the community string **comaccess** have read-only access to all objects. No other SNMP managers have access. SNMP Authentication Failure traps are sent by SNMPv2C to host `cisco.com` using community string **public**.

  ```
  Device(config)# snmp-server community comaccess ro 4
  Device(config)# snmp-server enable traps snmp authentication
  Device(config)# snmp-server host cisco.com version 2c public
  ```

- **Send Entity MIB traps to a restricted host**

  Enables the device to send Entity MIB traps to host `cisco.com`. The device restricts the use of the community string. The second command overwrites any previous **snmp-server host** commands for that host.

  ```
  Device(config)# snmp-server enable traps entity
  Device(config)# snmp-server host cisco.com restricted entity
  ```

- **Send all traps to a host**

  Enables the device to send all traps to host `myhost.cisco.com` using community string `public`.

```
Device(config)# snmp-server enable traps
Device(config)# snmp-server host myhost.cisco.com public
```

- **Configure SNMPv3 user with authentication and informs**

  Associates a user with a remote host and sends auth-level (authNoPriv) informs when the user enters global configuration mode.

```
Device(config)# snmp-server engineID remote 192.180.1.27
00000063000100a1c0b4011b
Device(config)# snmp-server group authgroup v3 auth
Device(config)# snmp-server user authuser authgroup remote 192.180.1.27 v3
auth md5 mypassword
Device(config)# snmp-server user authuser authgroup v3 auth md5 mypassword
Device(config)# snmp-server host 192.180.1.27 informs version 3 auth authuser
 config
Device(config)# snmp-server enable traps
Device(config)# snmp-server inform retries 0
```

- **Display SNMP manager polling statistics**

  Show entries of SNMP managers polled to an SNMP agent.

```
Device# show snmp stats host
Request Count                  Last Timestamp              Address
2                               00:00:01 ago               3.3.3.3
1                               1w2d ago                   2.2.2.2
```

- **SNMPv3 algorithm deprecation warnings**

  If you configure SNMPv3 users with the `md5`, `des`, or `3des` algorithms while compliance shield is disabled, the device displays warning messages indicating that these protocols will be deprecated.

```
Device(config)# snmp-server user md5user grp v3 auth md5 cisco1234 priv des
Sep  1 00:14:51.582 IST: %SNMP-6-AUTHPROTOCOLMD5: Authentication protocol
md5 support will be deprecated in future
Sep  1 00:14:51.582 IST: %SNMP-6-PRIVPROTOCOLDES: Privacy protocol des support
 will be deprecated in future
Sep  1 00:14:51.645 IST: %SNMP-5-WARMSTART: SNMP agent on host Switch is
undergoing a warm start
```

- **Compliance shield enabled warning for weaker algorithms**

  When compliance shield is enabled, the use of weaker algorithms `MD5`, `DES`, and `3DES` is not allowed for SNMP users, with a warning message.

```
Device(config)# snmp-server user md5user grp v3 auth md5 cisco1234
weaker algorithm MD5, DES and 3DES is not allowed for snmp user
```

Monitor SNMP status

This topic explains how to monitor SNMP status and display SNMP statistics using privileged EXEC commands.

To display SNMP input and output statistics, including the number of illegal community string entries, errors, and requested variables, use the **show snmp** privileged EXEC command.

You can use these privileged EXEC commands to display SNMP information.

Table 4: Commands to display SNMP status

Command	Purpose
show snmp	Displays SNMP statistics. Displays information on the local SNMP engine and all remote engines that have been configured on the device.
show snmp group	Displays information on each SNMP group on the network.
show snmp pending	Displays information on pending SNMP requests.
show snmp sessions	Displays information on the current SNMP sessions.
show snmp user	Displays information on each SNMP user name in the SNMP users table. **Note** You must use this command to display SNMPv3 configuration information for **auth | noauth | priv** mode. This information is not displayed in the **show running-config** output.

Note

By default, most IE switches have the PROFINET feature enabled. According to PROFINET specifications, if an interface requires an SFP module and the SFP module is not present, the SNMP `OID IF-MIB::ifOperStatus` reports the operational status as `notPresent(6)`. This status is returned when the SFP module is not plugged in, not detected, or is corrupted.