



Configuring SNMP

This chapter describes how to configure the Simple Network Management Protocol (SNMP) on the Catalyst 2970 switch.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the switch command reference for this release and to the *Cisco IOS Configuration Fundamentals Command Reference for Release 12.1*.

This chapter consists of these sections:

- [Understanding SNMP, page 21-1](#)
- [Configuring SNMP, page 21-5](#)
- [Displaying SNMP Status, page 21-12](#)

Understanding SNMP

SNMP is an application-layer protocol that provides a message format for communication between managers and agents. The SNMP system consists of an SNMP manager, an SNMP agent, and a management information base (MIB). The SNMP manager can be part of a network management system (NMS) such as CiscoWorks. The agent and MIB reside on the switch. To configure SNMP on the switch, you define the relationship between the manager and the agent.

The SNMP agent contains MIB variables whose values the SNMP manager can request or change. A manager can get a value from an agent or store a value into the agent. The agent gathers data from the MIB, the repository for information about device parameters and network data. The agent can also respond to a manager's requests to get or set data.

An agent can send unsolicited traps to the manager. Traps are messages alerting the SNMP manager to a condition on the network. Traps can mean improper user authentication, restarts, link status (up or down), MAC address tracking, closing of a Transmission Control Protocol (TCP) connection, loss of connection to a neighbor, or other significant events.

This section includes information about these topics:

- [SNMP Versions, page 21-2](#)
- [SNMP Manager Functions, page 21-3](#)
- [SNMP Agent Functions, page 21-3](#)
- [SNMP Community Strings, page 21-3](#)
- [Using SNMP to Access MIB Variables, page 21-4](#)
- [SNMP Notifications, page 21-4](#)

SNMP Versions

This software release supports these SNMP versions:

- **SNMPv1**—The Simple Network Management Protocol, a Full Internet Standard, defined in RFC 1157.
- **SNMPv2C** replaces the Party-based Administrative and Security Framework of SNMPv2Classic with the community-string-based Administrative Framework of SNMPv2C while retaining the bulk retrieval and improved error handling of SNMPv2Classic. It has these features:
 - **SNMPv2**—Version 2 of the Simple Network Management Protocol, a Draft Internet Standard, defined in RFCs 1902 through 1907.
 - **SNMPv2C**—The community-string-based Administrative Framework for SNMPv2, an Experimental Internet Protocol defined in RFC 1901.



Note

Though v3-related keywords are sometimes visible in the command-line help string, SNMPv3 is not supported in this release.

Both SNMPv1 and SNMPv2C use a community-based form of security. The community of managers able to access the agent's MIB is defined by an IP address access control list and password.

SNMPv2C includes a bulk retrieval mechanism and more detailed error message reporting to management stations. The bulk retrieval mechanism retrieves tables and large quantities of information, minimizing the number of round-trips required. The SNMPv2C improved error-handling includes expanded error codes that distinguish different kinds of error conditions; these conditions are reported through a single error code in SNMPv1. Error return codes in SNMPv2C report the error type.

You must configure the SNMP agent to use the SNMP version supported by the management station. Because an agent can communicate with multiple managers, you can configure the software to support communications using SNMPv1 and SNMPv2C protocols.

SNMP Manager Functions

The SNMP manager uses information in the MIB to perform the operations described in [Table 21-1](#).

Table 21-1 *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. ¹
get-bulk-request ²	Retrieves large blocks of data, such as multiple rows in a table, that would otherwise require the transmission of many small blocks of data.
get-response	Replies to a get-request, get-next-request, and set-request sent by an NMS.
set-request	Stores a value in a specific variable.
trap	An unsolicited message sent by an SNMP agent to an SNMP manager when some event has occurred.

1. 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.
2. The **get-bulk** command only works with SNMPv2 or later.

SNMP Agent Functions

The SNMP agent responds to SNMP manager requests as follows:

- **Get a MIB variable**—The SNMP agent begins this function in response to a request from the NMS. The agent retrieves the value of the requested MIB variable and responds to the NMS with that value.
- **Set a MIB variable**—The SNMP agent begins this function in response to a message from the NMS. The SNMP agent changes the value of the MIB variable to the value requested by the NMS.

The SNMP agent also sends unsolicited trap messages to notify an NMS that a significant event has occurred on the agent. Examples of trap conditions include, but are not limited to, when a port or module goes up or down, when spanning-tree topology changes occur, and when authentication failures occur.

SNMP Community Strings

SNMP community strings authenticate access to MIB objects and function as embedded passwords. In order for the NMS to access the switch, the community string definitions on the NMS must match at least one of the three community string definitions on the switch.

A community string can have one of these attributes:

- **Read-only (RO)**—Gives read access to authorized management stations to all objects in the MIB except the community strings, but does not allow write access
- **Read-write (RW)**—Gives read and write access to authorized management stations to all objects in the MIB, but does not allow access to the community strings
- **Read-write-all**—Gives read and write access to authorized management stations to all objects in the MIB, including the community strings

**Note**

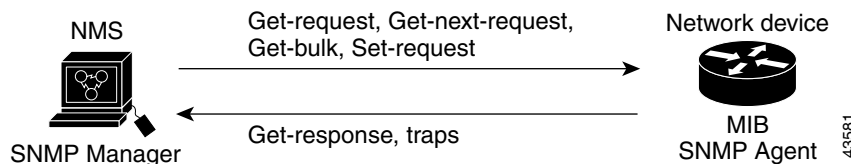
When a cluster is created, the command switch manages the exchange of messages among member switches and the SNMP application. The Cluster Management software appends the member switch number (@*esN*, where *N* is the switch number) to the first configured RW and RO community strings on the command switch and propagates them to the member switches. For more information, see [Chapter 5, “Clustering Switches.”](#)

Using SNMP to Access MIB Variables

An example of an NMS is the CiscoWorks network management software. CiscoWorks 2000 software uses the switch MIB variables to set device variables and to poll devices on the network for specific information. The results of a poll can be displayed as a graph and analyzed to troubleshoot internetworking problems, increase network performance, verify the configuration of devices, monitor traffic loads, and more.

As shown in [Figure 21-1](#), the SNMP agent gathers data from the MIB. The agent can send traps, or notification of certain events, to the SNMP manager, which receives and processes the traps. Traps alert the SNMP manager to a condition on the network such as improper user authentication, restarts, link status (up or down), MAC address tracking, and so forth. The SNMP agent also responds to MIB-related queries sent by the SNMP manager in *get-request*, *get-next-request*, and *set-request* format.

Figure 21-1 SNMP Network



For information on supported MIBs and how to access them, see [Appendix A, “Supported MIBs.”](#)

SNMP Notifications

SNMP allows the switch to send notifications to SNMP managers when particular events occur. SNMP notifications can be sent as traps or inform requests. In command syntax, unless there is an option in the command to select either traps or informs, the keyword *traps* refers to either traps or informs, or both. Use the **snmp-server host** command to specify whether to send SNMP notifications as traps or informs.

**Note**

SNMPv1 does not support informs.

Traps are unreliable because the receiver does not send an acknowledgment when it receives a trap, and the sender cannot determine if the trap was received. When an SNMP manager receives an inform request, it acknowledges the message with an SNMP response protocol data unit (PDU). If the sender does not receive a response, the inform request can be sent again. Because they can be re-sent, informs are more likely than traps to reach their intended destination.

The characteristics that make informs more reliable than traps also consume more resources in the switch and in the network. Unlike a trap, which is discarded as soon as it is sent, an inform request is held in memory until a response is received or the request times out. Traps are sent only once, but an inform might be re-sent or retried several times. The retries increase traffic and contribute to a higher overhead on the network. Therefore, traps and informs require a trade-off between reliability and resources. If it is important that the SNMP manager receive every notification, use inform requests. If traffic on the network or memory in the switch is a concern and notification is not required, use traps.

Configuring SNMP

This section describes how to configure SNMP on your switch. It contains this configuration information:

- [Default SNMP Configuration, page 21-5](#)
- [Disabling the SNMP Agent, page 21-6](#)
- [Configuring Community Strings, page 21-6](#)
- [Configuring SNMP Notifications, page 21-7](#)
- [Setting the Agent Contact and Location Information, page 21-10](#)
- [Limiting TFTP Servers Used Through SNMP, page 21-10](#)
- [SNMP Examples, page 21-11](#)

Default SNMP Configuration

[Table 21-2](#) shows the default SNMP configuration.

Table 21-2 Default SNMP Configuration

Feature	Default Setting
SNMP agent	Enabled
SNMP community strings	Read-Only: Public Read-Write: Private Read-Write-all: Secret
SNMP trap receiver	None configured
SNMP traps	None enabled
SNMP version	If no version keyword is present, the default is version 1.
SNMP notification type	If no type is specified, all notifications are sent.

Disabling the SNMP Agent

Beginning in privileged EXEC mode, follow these steps to disable the SNMP agent:

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	no snmp-server	Disable the SNMP agent operation.
Step 3	end	Return to privileged EXEC mode.
Step 4	show running-config	Verify your entries.
Step 5	copy running-config startup-config	(Optional) Save your entries in the configuration file.

The **no snmp-server** global configuration command disables all running versions (version 1 and version 2C) on the device. No specific IOS command exists to enable SNMP. The first **snmp-server** global configuration command that you enter enables all versions of SNMP.

Configuring Community Strings

You use the SNMP community string to define the relationship between the SNMP manager and the agent. The community string acts like a password to permit access to the agent on the switch. Optionally, you can specify one or more of these characteristics associated with the string:

- An access list of IP addresses of the SNMP managers that are permitted to use the community string to gain access to the agent
- A MIB view, which defines the subset of all MIB objects accessible to the given community
- Read and write or read-only permission for the MIB objects accessible to the community

Beginning in privileged EXEC mode, follow these steps to configure a community string on the switch:

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	snmp-server community <i>string</i> [view <i>view-name</i>] [ro rw] [<i>access-list-number</i>]	Configure the community string. <ul style="list-style-type: none"> • For <i>string</i>, specify a string that acts like a password and permits access to the SNMP protocol. You can configure one or more community strings of any length. • (Optional) For view, specify the view record accessible to the community. • (Optional) Specify either read-only (ro) if you want authorized management stations to retrieve MIB objects, or specify read-write (rw) if you want authorized management stations to retrieve and modify MIB objects. By default, the community string permits read-only access to all objects. • (Optional) For <i>access-list-number</i>, enter an IP standard access list numbered from 1 to 99 and 1300 to 1999.

	Command	Purpose
Step 3	access-list <i>access-list-number</i> { deny permit } <i>source</i> [<i>source-wildcard</i>]	<p>(Optional) If you specified an IP standard access list number in Step 2, then create the list, repeating the command as many times as necessary.</p> <ul style="list-style-type: none"> For <i>access-list-number</i>, enter the access list number specified in Step 2. The deny keyword denies access if the conditions are matched. The permit keyword permits access if the conditions are matched. For <i>source</i>, enter the IP address of the SNMP managers that are permitted to use the community string to gain access to the agent. (Optional) For <i>source-wildcard</i>, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. <p>Recall that the access list is always terminated by an implicit deny statement for everything.</p>
Step 4	end	Return to privileged EXEC mode.
Step 5	show running-config	Verify your entries.
Step 6	copy running-config startup-config	(Optional) Save your entries in the configuration file.

**Note**

To disable access for an SNMP community, set the community string for that community to the null string (do not enter a value for the community string).

To remove a specific community string, use the **no snmp-server community** *string* global configuration command.

This example shows how to assign the string *comaccess* to SNMP, to allow read-only access, and to specify that IP access list 4 can use the community string to gain access to the switch SNMP agent:

```
Switch(config)# snmp-server community comaccess ro 4
```

Configuring SNMP Notifications

A trap manager is a management station that receives and processes traps. Traps are system alerts that the switch generates when certain events occur. By default, no trap manager is defined, and no traps are sent. Switches running this IOS release can have an unlimited number of trap managers.

**Note**

Many commands use the word *traps* in the command syntax. Unless there is an option in the command to select either traps or informs, the keyword *traps* refers to either traps, informs, or both. Use the **snmp-server host** command to specify whether to send SNMP notifications as traps or informs.

[Table 21-3](#) describes the supported switch traps (notification types). You can enable any or all of these traps and configure a trap manager to receive them.

**Note**

Although visible in the command-line interface (CLI) online help, the **fru-ctrl** keyword is not supported.

Table 21-3 *Switch Notification Types*

Notification Type Keyword	Description
bridge	Generates STP bridge MIB traps.
cluster	Generates a trap when the cluster configuration changes.
config	Generates a trap for SNMP configuration changes.
entity	Generates a trap for SNMP entity changes.
envmon	Generates environmental monitor traps.
hsrp	Generates a trap for Hot Standby Router Protocol (HSRP) changes.
mac-notification	Generates a trap for MAC address notifications.
rtr	Generates a trap for the SNMP Response Time Reporter (RTR).
snmp	Generates a trap for SNMP-type notifications.
tty	Generates a trap for TCP connections.
vlan-membership	Generates a trap for SNMP VLAN membership changes.
vtp	Generates a trap for VLAN Trunking Protocol (VTP) changes.

You can use the **snmp-server host** global configuration command to a specific host to receive the notification types listed in [Table 21-3](#).

Beginning in privileged EXEC mode, follow these steps to configure the switch to send traps or informs to a host:

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	snmp-server host <i>host-addr</i> [traps informs] [version {1 2c}] <i>community-string</i> [udp-port <i>port</i>] [notification-type]	Specify the recipient of an SNMP trap operation. <ul style="list-style-type: none"> For <i>host-addr</i>, specify the name or Internet address of the host (the targeted recipient). (Optional) Enter traps (the default) to send SNMP traps to the host. (Optional) Enter informs to send SNMP informs to the host. (Optional) Specify the SNMP version (1 or 2c). SNMPv1 does not support informs. <p>Note Though visible in the command-line help string, the v3 keyword (SNMPv3) is not supported.</p> <ul style="list-style-type: none"> For <i>community-string</i>, enter the password-like community string sent with the notification operation. (Optional) For udp-port port, enter the remote device UDP port. (Optional) For <i>notification-type</i>, use the keywords listed in Table 21-3 on page 21-8. If no type is specified, all notifications are sent.
Step 3	snmp-server enable traps <i>notification-types</i>	Enable the switch to send traps or informs and specify the type of notifications to be sent. For a list of notification types, see Table 21-3 on page 21-8 , or enter this: snmp-server enable traps ? To enable multiple types of traps, you must enter a separate snmp-server enable traps command for each trap type. Although visible in the CLI online help, the fru-ctrl keyword is not supported.
Step 4	snmp-server trap-source <i>interface-id</i>	(Optional) Specify the source interface, which provides the IP address for the trap message. This command also sets the source IP address for informs.
Step 5	snmp-server queue-length <i>length</i>	(Optional) Establish the message queue length for each trap host. The range is 1 to 1000; the default is 10.
Step 6	snmp-server trap-timeout <i>seconds</i>	(Optional) Define how often to resend trap messages. The range is 1 to 1000; the default is 30 seconds.
Step 7	end	Return to privileged EXEC mode.
Step 8	show running-config	Verify your entries.
Step 9	copy running-config startup-config	(Optional) Save your entries in the configuration file.

The **snmp-server host** command specifies which hosts receive the notifications. The **snmp-server enable trap** command globally enables the mechanism for the specified notification (for traps and informs). To enable a host to receive an inform, you must configure an **snmp-server host informs** command for the host and globally enable informs by using the **snmp-server enable traps** command.

To remove the specified host from receiving traps, use the **no snmp-server host** *host* global configuration command. The **no snmp-server host** command with no keywords disables traps, but not informs, to the host. To disable informs, use the **no snmp-server host informs** global configuration command. To disable a specific trap type, use the **no snmp-server enable traps** *notification-types* global configuration command.

Setting the Agent Contact and Location Information

Beginning in privileged EXEC mode, follow these steps to set the system contact and location of the SNMP agent so that these descriptions can be accessed through the configuration file:

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	snmp-server contact <i>text</i>	Set the system contact string. For example: snmp-server contact Dial System Operator at beeper 21555.
Step 3	snmp-server location <i>text</i>	Set the system location string. For example: snmp-server location Building 3/Room 222
Step 4	end	Return to privileged EXEC mode.
Step 5	show running-config	Verify your entries.
Step 6	copy running-config startup-config	(Optional) Save your entries in the configuration file.

Limiting TFTP Servers Used Through SNMP

Beginning in privileged EXEC mode, 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:

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	snmp-server tftp-server-list <i>access-list-number</i>	Limit TFTP servers used for configuration file copies through SNMP to the servers in the access list. For <i>access-list-number</i> , enter an IP standard access list numbered from 1 to 99 and 1300 to 1999.

	Command	Purpose
Step 3	<code>access-list access-list-number {deny permit} source [source-wildcard]</code>	<p>Create a standard access list, repeating the command as many times as necessary.</p> <ul style="list-style-type: none"> For <i>access-list-number</i>, enter the access list number specified in Step 2. The deny keyword denies access if the conditions are matched. The permit keyword permits access if the conditions are matched. For <i>source</i>, enter the IP address of the TFTP servers that can access the switch. (Optional) For <i>source-wildcard</i>, enter the wildcard bits, in dotted decimal notation, to be applied to the source. Place ones in the bit positions that you want to ignore. <p>Recall that the access list is always terminated by an implicit deny statement for everything.</p>
Step 4	<code>end</code>	Return to privileged EXEC mode.
Step 5	<code>show running-config</code>	Verify your entries.
Step 6	<code>copy running-config startup-config</code>	(Optional) Save your entries in the configuration file.

SNMP Examples

This example shows how to enable all versions of SNMP. The configuration permits any SNMP manager to access all objects with read-only permissions using the community string *public*. This configuration does not cause the switch to send any traps.

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

This example shows how to permit any SNMP manager to access all objects with read-only permission using the community string *public*. The switch also sends VTP traps to the hosts 192.180.1.111 and 192.180.1.33 using SNMPv1 and to the host 192.180.1.27 using SNMPv2C. The community string *public* is sent with the traps.

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

This example shows how to allow read-only access for all objects to members of access list 4 that use the *comaccess* community string. No other SNMP managers have access to any objects. SNMP Authentication Failure traps are sent by SNMPv2C to the host *cisco.com* using the community string *public*.

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

This example shows how to send Entity MIB traps to the host *cisco.com*. The community string is restricted. The first line enables the switch to send Entity MIB traps in addition to any traps previously enabled. The second line specifies the destination of these traps and overwrites any previous **snmp-server host** commands for the host *cisco.com*.

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

This example shows how to enable the switch to send all traps to the host *myhost.cisco.com* using the community string *public*:

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

Displaying SNMP Status

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. For information about the fields in the output displays, refer to the *Cisco IOS Configuration Fundamentals Command Reference for Release 12.1*.