



Simple Network Management Protocol

This document describes how to configure the Simple Network Management Protocol (SNMP).

This document consists of these sections:

- [Understanding SNMP, page 1](#)
- [Configuring SNMP, page 4](#)
- [Displaying SNMP Status, page 10](#)

Understanding SNMP

SNMP is an application-layer protocol that provides a message format for communication between SNMP managers and agents. The SNMP manager can be part of a network management system (NMS) such as CiscoWorks. The agent and management information base (MIB) reside on the WMIC. To configure SNMP on the WMIC, 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 TCP connection, loss of connection to a neighbor, or other significant events.

This section includes these concepts:

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



Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2008 Cisco Systems, Inc. All rights reserved.

SNMP Versions

This software release supports these SNMP versions:

- SNMPv1—The Simple Network Management Protocol, a full Internet standard, defined in RFC 1157.
- SNMPv2C, which 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-based Administrative Framework for SNMPv2, an experimental Internet protocol defined in RFC 1901.

SNMPv2C replaces the Party-based Administrative and Security Framework of SNMPv2Classic with the Community-based Administrative Framework of SNMPv2C while retaining the bulk retrieval and improved error handling of SNMPv2Classic.

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 now report the error type.

You must configure the SNMP agent to use the version of SNMP supported by the management station. An agent can communicate with multiple managers; therefore, you can configure the software to support communications with one management station using the SNMPv1 protocol and another using the SNMPv2 protocol.

SNMP Manager Functions

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

Table 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 that would otherwise require the transmission of many small blocks of data, such as multiple rows in a table. |
| 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 works only with SNMPv2.

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 WMIC, the community string definitions on the NMS must match at least one of the three community string definitions on the WMIC.

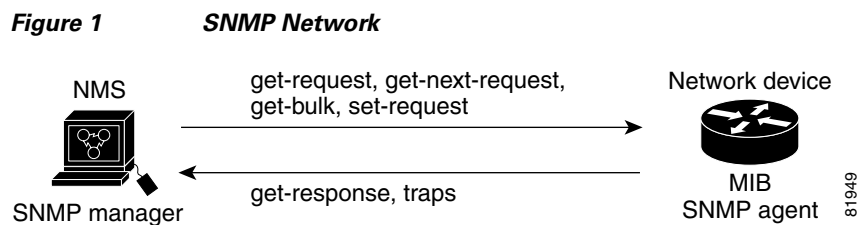
A community string can have one of these attributes:

- Read-only—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—Gives read and write access to authorized management stations to all objects in the MIB, but does not allow access to the community strings

Using SNMP to Access MIB Variables

An example of an NMS is the CiscoWorks network management software. CiscoWorks 2000 software uses the 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 1](#), the SNMP agent gathers data from the MIB. The agent can send traps (notification of certain events) to the SNMP manager, which receives and processes the traps. Traps are messages alerting 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.



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

Configuring SNMP

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

- [Default SNMP Configuration, page 4](#)
- [Enabling the SNMP Agent, page 4](#)
- [Configuring Community Strings, page 4](#)
- [Configuring Trap Managers and Enabling Traps, page 6](#)
- [Setting the Agent Contact and Location Information, page 9](#)
- [Using the snmp-server view Command, page 9](#)
- [SNMP Examples, page 9](#)

Default SNMP Configuration

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

Table 2 *Default SNMP Configuration*

| Feature | Default Setting |
|------------------------|-----------------|
| SNMP agent | Disabled |
| SNMP community strings | None configured |
| SNMP trap receiver | None configured |
| SNMP traps | None enabled |

Enabling the SNMP Agent

No specific IOS command exists to enable SNMP. The first **snmp-server** global configuration command that you enter enables SNMPv1 and SNMPv2.

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 WMIC.

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

**Note**

In the current Cisco IOS MIB agent implementation, the default community string is for the Internet MIB object sub-tree. Because IEEE802dot11 is under another branch of the MIB object tree, you must enable either a separate community string and view on the IEEE802dot11 MIB or a common view and community string on the ISO object in the MIB object tree. ISO is the common parent node of IEEE (IEEE802dot11) and Internet. This MIB agent behavior is different from the MIB agent behavior on access points not running IOS software.

To configure a community string on the WMIC, follow these steps, beginning in privileged EXEC mode:

| | Command | Purpose |
|--------|---|---|
| Step 1 | configure terminal | Enters global configuration mode. |
| Step 2 | snmp-server community <i>string</i> [<i>access-list-number</i>] [view <i>mib-view</i>] [ro rw] | <p>Configures the community string.</p> <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 <i>access-list-number</i>, enter an IP standard access list numbered from 1 to 99 and 1300 to 1999. (Optional) For view <i>mib-view</i>, specify a MIB view to which this community has access, such as ieee802dot11. See the “Using the snmp-server view Command” section on page 9 for instructions on using the snmp-server view command to access Standard IEEE 802.11 MIB objects through IEEE view. (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. <p>Note To access the IEEE 802dot11 MIB, you must enable either a separate community string and view on the IEEE802dot11 MIB or a common view and community string on the ISO object in the MIB object tree.</p> |

| | Command | Purpose |
|--------|---|---|
| Step 3 | <code>access-list <i>access-list-number</i> {deny permit} <i>source</i> [<i>source-wildcard</i>]</code> | <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 | <code>end</code> | Returns to privileged EXEC mode. |
| Step 5 | <code>show running-config</code> | Verifies your entries. |
| Step 6 | <code>copy running-config startup-config</code> | (Optional) Saves your entries in the configuration file. |

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 strings *open* and *ieee* to SNMP, to allow read-write access for both, and to specify that *open* is the community string for queries on non-IEEE802dot11-MIB objects and *ieee* is the community string for queries on IEEE802dot11-mib objects:

```
bridge(config)# snmp-server view dot11view ieee802dot11 included
bridge(config)# snmp-server community open rw
bridge(config)# snmp-server community ieee view ieee802dot11 rw
```

Configuring Trap Managers and Enabling Traps

A trap manager is a management station that receives and processes traps. Traps are system alerts that the device generates when certain events occur. By default, no trap manager is defined, and no traps are issued.

Bridges running this IOS release can have an unlimited number of trap managers. Community strings can be any length.

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

Table 3 Notification Types

| Notification Type | Description |
|---------------------------|---|
| authenticate-fail | Enables traps for authentication failures. |
| config | Enables traps for SNMP configuration changes. |
| deauthenticate | Enables traps for client device deauthentications. |
| disassociate | Enables traps for client device disassociations. |
| dot11-qos | Enables traps for QoS changes. |
| entity | Enables traps for SNMP entity changes. |
| envmon temperature | Enables traps for monitoring radio temperature. This trap is sent out when the WMIC radio temperature approaches the limits of its operating range. |
| snmp | Enables traps for SNMP events. |
| syslog | Enables syslog traps. |
| wlan-wep | Enables WEP traps. |

Some notification types cannot be controlled with the **snmp-server enable** global configuration command, such as **tty** and **udp-port**. These notification types are always enabled. You can use the **snmp-server host** global configuration command to a specific host to receive the notification types listed in [Table 3](#).

To configure the WMIC to send traps to a host, follow these steps, beginning in privileged EXEC mode:

| | Command | Purpose |
|--------|---|---|
| Step 1 | configure terminal | Enters global configuration mode. |
| Step 2 | snmp-server host <i>host-addr</i> { traps informs } { version { 1 2c }} <i>community-string notification-type</i> | <p>Specifies the recipient of the trap message.</p> <ul style="list-style-type: none"> For <i>host-addr</i>, specify the name or address of the host (the targeted recipient). Specify traps (the default) to send SNMP traps to the host. Specify informs to send SNMP informs to the host. Specify the SNMP version to support. Version 1, the default, is not available with informs. <p>Note Though visible in the command-line help string, the version 3 keyword (SNMPv3) is not supported.</p> <ul style="list-style-type: none"> For <i>community-string</i>, specify the string to send with the notification operation. Though you can set this string using the snmp-server host command, we recommend that you define this string by using the snmp-server community command before using the snmp-server host command. For <i>notification-type</i>, use the keywords listed in Table 3 on page 7. |
| Step 3 | snmp-server enable traps <i>notification-types</i> | <p>Enables the WMIC to send specific traps. For a list of traps, see Table 3 on page 7.</p> <p>To enable multiple types of traps, you must issue a separate snmp-server enable traps command for each trap type.</p> |
| Step 4 | end | Returns to privileged EXEC mode. |
| Step 5 | show running-config | Verifies your entries. |
| Step 6 | copy running-config startup-config | (Optional) Saves your entries in the configuration file. |

To remove the specified host from receiving traps, use the **no snmp-server host** *host* 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

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

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

Using the snmp-server view Command

In global configuration mode, use the **snmp-server view** command to access Standard IEEE 802.11 MIB objects through IEEE view and the dot11 read-write community string.

This example shows how to enable IEEE view and dot11 read-write community string:

```
bridge(config)# snmp-server view ieee ieee802dot11 included
bridge(config)# snmp-server community dot11 view ieee RW
```

SNMP Examples

The following example shows how to enable SNMPv1 and SNMPv2C. 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 WMIC to send any traps.

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

The following example shows how to assign the strings *open* and *ieee* to SNMP, to allow read-write access for both, and to specify that *open* is the community string for queries on non-IEEE802dot11-MIB objects and *ieee* is the community string for queries on IEEE802dot11-mib objects:

```
bridge(config)# snmp-server view dot11view ieee802dot11 included
bridge(config)# snmp-server community open rw
bridge(config)# snmp-server community ieee view ieee802dot11 rw
```

The following example shows how to permit any SNMP manager to access all objects with read-only permission using the community string *public*. The WMIC also sends config 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.

```
bridge(config)# snmp-server community public
bridge(config)# snmp-server enable traps config
```

```
bridge(config)# snmp-server host 192.180.1.27 version 2c public
bridge(config)# snmp-server host 192.180.1.111 version 1 public
bridge(config)# snmp-server host 192.180.1.33 public
```

The following 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*.

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

The following example shows how to send Entity MIB traps to the host *cisco.com*. The community string is restricted. The first line enables the WMIC 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*.

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

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

```
bridge(config)# snmp-server enable traps
bridge(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 this display, refer to the *Cisco IOS Configuration Fundamentals Command Reference for Release 12.2*.