



## Configuring SNMP

---

This chapter describes how to configure the Simple Network Management Protocol (SNMP) on your access point.



**Note**

---

For complete syntax and usage information for the commands used in this chapter, refer to the *Cisco IOS Command Reference for Cisco Aironet Access Points* for this release.

---

# 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 Cisco Prime Infrastructure. The agent and management information base (MIB) reside on the access point. To configure SNMP on the access point, 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 18-2](#)
- [SNMP Manager Functions, page 18-3](#)
- [SNMP Agent Functions, page 18-4](#)
- [SNMP Community Strings, page 18-4](#)
- [Using SNMP to Access MIB Variables, page 18-4](#)

## SNMP Versions

This software release supports these SNMP versions:

- SNMPv1—The Simple Network Management Protocol, a full Internet standard, defined in RFC 1157.
- SNMPv2C, 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.
- SNMPv3, which has these features:
  - Support for SHA and MD5 authentication protocols and DES56 encryption.
  - Three security levels: no authentication and no privacy (NoAuthNoPriv), authentication and no privacy (AuthNoPriv), and authentication and privacy (AuthPriv).

SNMPv3 supports the highest available levels of security for SNMP communication. Community strings for SNMPv1 and SNMPv2 are stored and transferred as plain text without encryption. In the SNMPv3 security model, SNMP users authenticate and join a user group. Access to system data is restricted based on the group.

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 SNMPv3 protocol and another using the SNMPv2 or SNMPv1 protocol.

Table 18-1 lists the SNMP versions and security levels supported on access points.

**Table 18-1** *SNMP Versions and Security Levels*

SNMP Version	Security Level	Authentication	Encryption
v1	NoAuthNoPriv	Community string match	None
v2C	NoAuthNoPriv	Community string match	None
v3	NoAuthNoPriv	Username match	None
v3	AuthNoPriv	HMAC-MD5 or HMAC-SHA algorithms	None
v3	AuthPriv	HMAC-MD5 or HMAC-SHA algorithms	DES 56-bit encryption

## SNMP Manager Functions

The SNMP manager uses information in the MIB to perform the operations described in Table 18-2.

**Table 18-2** *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. <sup>1</sup>
get-bulk-request <sup>2</sup>	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 access point, the community string definitions on the NMS must match at least one of the three community string definitions on the access point.

**Note**

---

SNMP communities are used with SNMPv1 and SNMPv2c. SNMPv3 does not use communities.

---

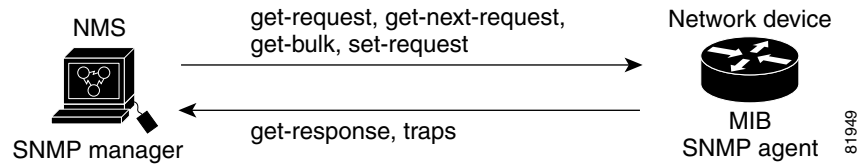
A community string can have one of these attributes:

- 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 Cisco Prime Infrastructure network management software. Cisco Prime Infrastructure software uses the access point MIB variables to set device variables and to poll devices on the network for specific information. The results of a poll can be displayed and analyzed to troubleshoot internetworking problems, increase network performance, verify the configuration of devices, monitor traffic loads, and more.

As shown in [Figure 18-1](#), 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.

**Figure 18-1** SNMP Network

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

## Configuring SNMP

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

- [Default SNMP Configuration, page 18-5](#)
- [Enabling the SNMP Agent, page 18-6](#)
- [Configuring Community Strings, page 18-6](#)
- [Specifying SNMP-Server Group Names, page 18-8](#)
- [Configuring SNMP-Server Hosts, page 18-8](#)
- [Configuring SNMP-Server Users, page 18-8](#)
- [Configuring Trap Managers and Enabling Traps, page 18-8](#)
- [Setting the Agent Contact and Location Information, page 18-11](#)
- [Using the snmp-server view Command, page 18-11](#)
- [SNMP Examples, page 18-12](#)

## Default SNMP Configuration

[Table 18-3](#) shows the default SNMP configuration.

**Table 18-3** Default SNMP Configuration

Feature	Default Setting
SNMP agent	Disabled
SNMP community strings	No strings are configured by default. However, when you enable SNMP using the web-browser interface, the access point automatically creates the <i>public</i> community with read-only access to the IEEE802dot11 MIB.
SNMP trap receiver	None configured
SNMP traps	None enabled

## Enabling the SNMP Agent

No specific CLI command exists to enable SNMP. The first **snmp-server** global configuration command that you enter enables the supported versions of SNMP.

You can also enable SNMP on the SNMP Properties page on the web-browser interface. When you enable SNMP on the web-browser interface, the access point automatically creates a community string called *public* with read-only access to the IEEE802dot11 MIB.

## 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 access point.

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 Cisco IOS software.

---

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

	Command	Purpose
Step 1	<b>configure terminal</b>	Enter global configuration mode.
Step 2	<b>snmp-server community</b> <i>string</i> [ <i>access-list-number</i> ] [ <b>view</b> <i>mib-view</i> ] [ <b>ro</b>   <b>rw</b> ]	<p>Configure the community string.</p> <ul style="list-style-type: none"> <li>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.</li> <li>(Optional) For <i>access-list-number</i>, enter an IP standard access list numbered from 1 to 99 and 1300 to 1999.</li> <li>(Optional) For <b>view</b> <i>mib-view</i>, specify a MIB view to which this community has access, such as <b>ieee802dot11</b>. See the “Using the <b>snmp-server view</b> Command” section on page 18-11 for instructions on using the <b>snmp-server view</b> command to access Standard IEEE 802.11 MIB objects through IEEE view.</li> <li>(Optional) Specify either read-only (<b>ro</b>) if you want authorized management stations to retrieve MIB objects, or specify read/write (<b>rw</b>) if you want authorized management stations to retrieve and modify MIB objects. By default, the community string permits read-only access to all objects.</li> </ul> <p><b>Note</b> To access the IEEE802dot11 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>
Step 3	<b>access-list</b> <i>access-list-number</i> { <b>deny</b>   <b>permit</b> } <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"> <li>For <i>access-list-number</i>, enter the access list number specified in Step 2.</li> <li>The <b>deny</b> keyword denies access if the conditions are matched. The <b>permit</b> keyword permits access if the conditions are matched.</li> <li>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.</li> <li>(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.</li> </ul> <p>Recall that the access list is always terminated by an implicit deny statement for everything.</p>
Step 4	<b>end</b>	Return to privileged EXEC mode.
Step 5	<b>show running-config</b>	Verify your entries.
Step 6	<b>copy running-config startup-config</b>	(Optional) Save 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 all objects:

```
ap(config)# snmp-server community open rw
ap(config)# snmp-server community ieee view ieee802dot11 rw
```

## Specifying SNMP-Server Group Names

To configure a new SNMP group, or a table that maps SNMP users to SNMP views, use the following command in global configuration mode:

Command	Purpose
<b>snmp-server group</b> [ <i>groupname</i> { <b>v1</b>   <b>v2c</b>   <b>v3</b> [ <b>auth</b>   <b>noauth</b>   <b>priv</b> ]}][ <b>read</b> <i>readview</i> ] [ <b>write</b> <i>writeview</i> ] [ <b>notify</b> <i>notifyview</i> ] [ <b>access</b> <i>access-list</i> ]	Configures a new SNMP group, or a table that maps SNMP users to SNMP views.

## Configuring SNMP-Server Hosts

To configure the recipient of an SNMP trap operation, use the following command in global configuration mode:

Command	Purpose
<b>snmp-server host</b> <i>host</i> [ <b>traps</b>   <b>informs</b> ][ <b>version</b> { <b>1</b>   <b>2c</b>   <b>3</b> [ <b>auth</b>   <b>noauth</b>   <b>priv</b> ]}] <i>community-string</i> [ <b>udp-port</b> <i>port</i> ] [ <i>notification-type</i> ]	Configures the recipient of an SNMP trap operation.

## Configuring SNMP-Server Users

To configure a new user to an SNMP group, use the following command in global configuration mode:

Command	Purpose
<b>snmp-server user</b> <i>username</i> [ <i>groupname</i> <b>remote</b> <i>ip-address</i> ] [ <b>udp-port</b> <i>port</i> ] { <b>v1</b>   <b>v2c</b>   <b>v3</b> [ <b>encrypted</b> ] [ <b>auth</b> { <b>md5</b>   <b>sha</b> }] <i>auth-password</i> [ <b>priv</b> <b>des56</b> <i>priv password</i> ]} [ <b>access</b> <i>access-list</i> ]	Configures a new user to an SNMP group.

## Configuring Trap Managers and Enabling Traps

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



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

Table 18-4 describes the supported access point traps (notification types). You can enable any or all of these traps and configure a trap manager to receive them.

**Table 18-4 Notification Types**

Notification Type	Description
<b>aaa_server</b>	Enable traps for AAA events
<b>associate</b>	Enable traps for client device associations <sup>1</sup> .
<b>authenticate-fail</b>	Enable traps for authentication failures.
<b>config</b>	Enable traps for SNMP configuration changes.
<b>deauthenticate</b>	Enable traps for client device deauthentications.
<b>disassociate</b>	Enable traps for client device disassociations.
<b>dot11-qos</b>	Enable traps for QoS changes.
<b>entity</b>	Enable traps for SNMP entity changes.
<b>rogue-ap</b>	Enable traps for rogue access point detections.
<b>snmp</b>	Enable traps for SNMP events.
<b>switch-over</b>	Enable traps for switch-overs.
<b>syslog</b>	Enable syslog traps.
<b>wlan-wep</b>	Enable WEP traps.
<b>cef</b>	Allows cef traps
<b>config-copy</b>	Allow SNMP config-copy traps
<b>config-ctid</b>	Allow SNMP config-ctid traps
<b>cpu</b>	Allow cpu related traps
<b>dot11-mibs</b>	Allow dot11 traps <sup>2</sup> : <ul style="list-style-type: none"> <li>• <b>associate</b>—Enable traps for client device associations.</li> <li>• <b>authenticate-fail</b>—Enable traps for authentication failures.</li> <li>• <b>deauthenticate</b>—Enable traps for client device deauthentications.</li> <li>• <b>disassociate</b>—Enable traps for client device disassociations.</li> <li>• <b>qos</b>—Enable traps for QoS changes.</li> <li>• <b>rogue-ap</b>—Enable traps for rogue access point detections.</li> <li>• <b>switch-over</b>—Enable traps for switch-overs.</li> <li>• <b>wlan-wep</b>—Enable WEP traps.</li> </ul>
<b>entity</b>	Allow SNMP entity traps
<b>l2tun-pseudowire-status</b>	Allow SNMP L2 pseudowire status traps
<b>l2tun-session</b>	Allow SNMP L2 session traps
<b>syslog</b>	Allow SNMP syslog traps
<b>tty</b>	Allow TCP connection traps

**Table 18-4 Notification Types (continued)**

Notification Type	Description
<b>udp-port</b>	The notification host's UDP port number
<b>vrfmib</b>	Allow SNMP vrfmib traps

1. The association trap is only supported on IW3702. If you want to enable the association trap from web GUI, you can only configure it through “Enable All Trap Notifications.”
2. For each dot11 trap, there is an equivalent notification type. For example, **snmp-server enable traps dot11-mibs qos** equals to **snmp-server enable traps dot11-qos**. The output of **show running-config** command will not display dot11 traps but only the equivalent notification types.

Some notification types cannot be controlled with the **snmp-server enable** global configuration command, such as **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 18-4](#).

Beginning in privileged EXEC mode, follow these steps to configure the access point to send traps to a host:

	Command	Purpose
<b>Step 1</b>	<b>configure terminal</b>	Enter global configuration mode.
<b>Step 2</b>	<b>snmp-server host</b> <i>host-addr</i> { <b>traps</b>   <b>informs</b> } { <b>version</b> { <b>1</b>   <b>2c</b>   <b>3</b> { <b>auth</b>   <b>noauth</b>   <b>priv</b> }}} <i>community-string</i> [ <b>udp-port</b> <i>port</i> ] <i>notification-type</i>	Specify the recipient of the trap message. <ul style="list-style-type: none"> <li>• For <i>host-addr</i>, specify the name or address of the host (the targeted recipient).</li> <li>• Specify <b>traps</b> (the default) to send SNMP traps to the host. Specify <b>informs</b> to send SNMP informs to the host.</li> <li>• Specify the SNMP version to support. Version 1, the default, is not available with informs. Version 3 has three security levels: <ul style="list-style-type: none"> <li>– <b>auth</b>—Specifies authentication of packets without encryption</li> <li>– <b>noauth</b>—Specifies no authentication and no encryption for packets</li> <li>– <b>priv</b>—Specifies authentication and encryption for packets</li> </ul> </li> <li>• For <i>community-string</i>, specify the string to send with the notification operation. Though you can set this string using the <b>snmp-server host</b> command, We recommend that you define this string by using the <b>snmp-server community</b> command before using the <b>snmp-server host</b> command.</li> <li>• For <i>notification-type</i>, use the keywords listed in <a href="#">Table 18-4</a> on page 18-9.</li> </ul>

	Command	Purpose
Step 3	<code>snmp-server enable traps notification-types</code>	Enable the access point to send specific traps. For a list of traps, see <a href="#">Table 18-4 on page 18-9</a> .  To enable multiple types of traps, you must issue a separate <code>snmp-server enable traps</code> command for each trap type.
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.

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

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	<code>configure terminal</code>	Enter global configuration mode.
Step 2	<code>snmp-server contact text</code>	Set the system contact string.  For example: <code>snmp-server contact Dial System Operator at beeper 21555.</code>
Step 3	<code>snmp-server location text</code>	Set the system location string.  For example: <code>snmp-server location Building 3/Room 222</code>
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.

## 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:

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

## SNMP Examples

This example shows how to enable SNMPv1, SNMPv2C, and SNMPv3. 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 access point to send any traps.

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

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
```

This example shows how to permit any SNMP manager to access all objects with read-only permission using the community string *public*. The access point 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.

```
AP(config)# snmp-server community public
AP(config)# snmp-server enable traps config
AP(config)# snmp-server host 192.180.1.27 version 2c public
AP(config)# snmp-server host 192.180.1.111 version 1 public
AP(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*.

```
AP(config)# snmp-server community comaccess ro 4
AP(config)# snmp-server enable traps snmp authentication
AP(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 access point 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*.

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

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

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

This example shows how to enable the access point to send association traps with wireless client Mac address and autonomous AP IP address:

```
AP(config)# snmp-server enable traps dot11-mibs associate
```

This example shows how to enable the access point to send disassociation traps with wireless client Mac address:

```
AP(config)# snmp-server enable traps dot11-mibs disassociate
```

**Note**

The association and disassociation traps only apply for wireless clients. The trap will be triggered when a wireless client associates or disassociates to the autonomous AP. Wired client will not trigger these two traps. By default, both the association and disassociation traps are disabled.

This example shows how to configure these SNMPv3 settings:

- a view name (*iso*)
- an SNMP engine ID (*1234567890*) that this agent uses to identify itself to the remote host at IP address *1.4.74.10*
- an SNMPv3 group (*admin*) which supports privacy encryption, and all users of the group have read and write access to all objects defined in the *iso* view
- an SNMP user (*joe*) that belongs to the admin group, uses MD5 authentication for queries, uses *xyz123* as a password for MD5, uses DES56 data query encryption, and uses *key007* as an encryption key
- an SNMP user (*fred*) that belongs to the admin group, uses MD5 authentication for queries, uses *abc789* as an encrypted password for MD5, uses DES56 data query encryption, and uses *key99* as an encryption key

```
AP(config)# snmp-server view iso iso included
AP(config)# snmp-server engineID remote 1.4.74.10 1234567890
AP(config)# snmp-server group admin v3 priv
AP(config)# snmp-server group admin v3 priv read iso write iso
AP(config)# snmp-server user joe admin v3 auth md5 xyz123 priv des56 key007
AP(config)# snmp-server user fred admin v3 encrypted auth md5 abc789 priv des56 key99
```

**Note**

After you enter the last command in this example, the **show running-config** and **show startup-config** commands display only a partial SNMP configuration.

## 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*.

