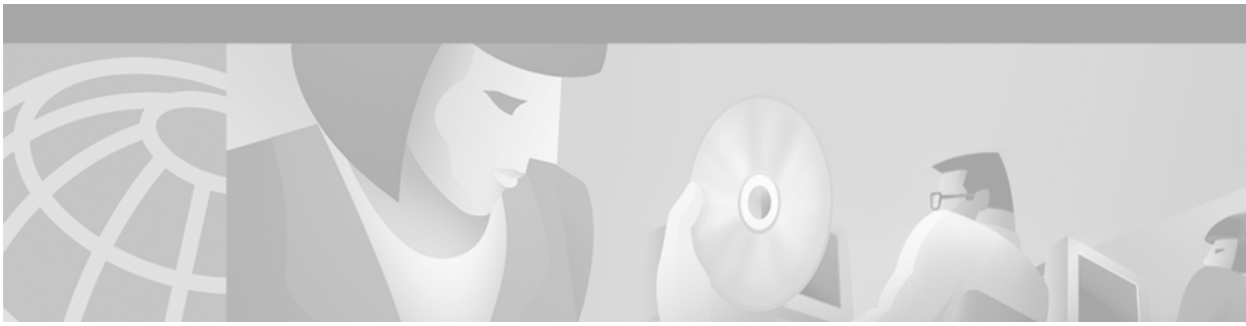


Configuring SNMP Support

This chapter describes the Simple Network Management Protocol (SNMP), SNMP MIBs, and how to configure SNMP on Cisco devices.

For a complete description of the router monitoring commands mentioned in this chapter, see the "SNMP Commands" chapter in the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*. To locate documentation of other commands that appear in this chapter, use the *Cisco IOS Command Reference Master Index* or search online. For further information about using SNMP, see the SNMP Technical Tips area on Cisco.com at http://www.cisco.com/warp/public/477/SNMP/snmp-indx.html.

To identify hardware or software image support for a specific feature, use Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Platform Support for Cisco IOS Software Features" section in the "About Cisco IOS Software Documentation" chapter.

This chapter contains the following sections:

- Understanding SNMP
- SNMP Configuration Task List
- SNMP Configuration Examples
- New MIB Features in Cisco IOS Release 12.2

Understanding SNMP

SNMP is an application-layer protocol that provides a message format for communication between SNMP managers and agents. SNMP provides a standardized framework and a common language used for the monitoring and management of devices in a network.

The SNMP framework has three parts:

- An SNMP manager
- An SNMP agent
- A MIB

The SNMP manager is the system used to control and monitor the activities of network hosts using SNMP. The most common managing system is called a Network Management System (NMS). The term NMS can be applied to either a dedicated device used for network management, or the applications used on such a device. A variety of network management applications are available for use with SNMP. These features range from simple command-line applications to feature-rich graphical user interfaces (such as the CiscoWorks2000 line of products).

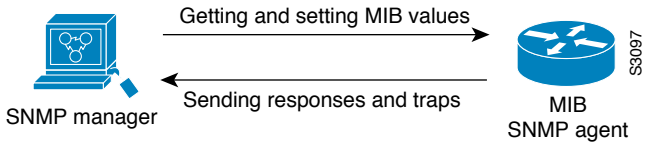
The SNMP agent is the software component within the managed device that maintains the data for the device and reports these data, as needed, to managing systems. The agent and MIB reside on the routing device (router, access server, or switch). To enable the SNMP agent on a Cisco routing device, you must define the relationship between the manager and the agent.

The Management Information Base (MIB) is a virtual information storage area for network management information, which consists of collections of managed objects. Within the MIB there are collections of related objects, defined in MIB modules. MIB modules are written in the SNMP MIB module language, as defined in STD 58, RFC 2578, RFC 2579, and RFC 2580 (see the "MIBs and RFCs" section for an explanation of RFC and STD documents). Note that individual MIB modules are also referred to as MIBs; for example, the Interfaces Group MIB (IF-MIB) is a MIB module within *the* MIB on your system.

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

Figure 14 illustrates the communications relationship between the SNMP manager and agent. A manager can send the agent requests to get and set MIB values. The agent can respond to these requests. Independent of this interaction, the agent can send unsolicited notifications (traps or informs) to the manager to notify the manager of network conditions.

Figure 14 Communication Between an SNMP Agent and Manager



Note This chapter discusses how to enable the SNMP agent on your Cisco device, and how to control the sending of SNMP notifications from the agent. For information on using SNMP management systems, see the appropriate documentation for your NMS application.

SNMP Notifications

A key feature of SNMP is the ability to generate notifications from an SNMP agent. These notifications do not require that requests be sent from the SNMP manager. Unsolicited (asynchronous) notifications can be generated as *traps* or *inform requests*. Traps are messages alerting the SNMP manager to a condition on the network. Inform requests (informs) are traps that include a request for confirmation of receipt from the SNMP manager. Notifications can indicate improper user authentication, restarts, the closing of a connection, loss of connection to a neighbor router, or other significant events.

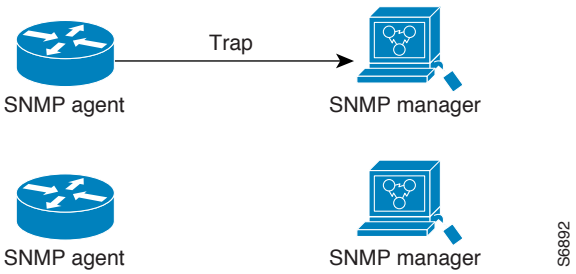
Traps are less reliable than informs because the receiver does not send any acknowledgment when it receives a trap. The sender cannot determine if the trap was received. An SNMP manager that receives an inform request acknowledges the message with an SNMP response protocol data unit (PDU). If the manager does not receive an inform request, it does not send a response. If the sender never receives a response, the inform request can be sent again. Thus, informs are more likely to reach their intended destination.

However, traps are often preferred because informs consume more resources in the router and in the network. Unlike a trap, which is discarded as soon as it is sent, an inform request must be held in memory until a response is received or the request times out. Also, traps are sent only once, while an inform may be retried several times. The retries increase traffic and contribute to a higher overhead on the network. Thus, traps and inform requests provide a trade-off between reliability and resources. If it is important that the SNMP manager receives every notification, use inform requests. However, if you are concerned about traffic on your network or memory in the router and you need not receive every notification, use traps.

Figure 15 through Figure 18 illustrate the differences between traps and inform requests.

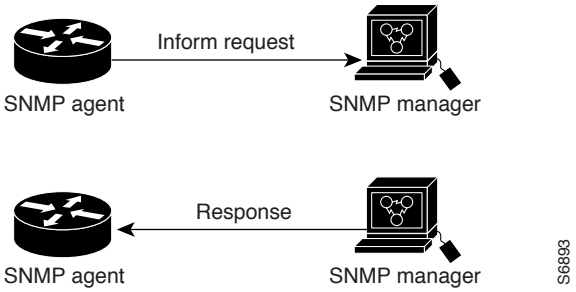
In Figure 15, the agent router successfully sends a trap to the SNMP manager. Although the manager receives the trap, it does not send any acknowledgment to the agent. The agent has no way of knowing that the trap reached its destination.

Figure 15 Trap Successfully Sent to SNMP Manager



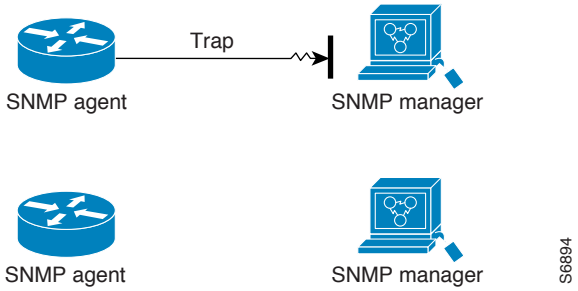
In Figure 16, the agent router successfully sends an inform request to the manager. When the manager receives the inform request, it sends a response to the agent. Thus, the agent knows that the inform request reached its destination. Notice that, in this example, twice as much traffic is generated as in Figure 15; however, the agent knows that the manager received the notification.

Figure 16 Inform Request Successfully Sent to SNMP Manager



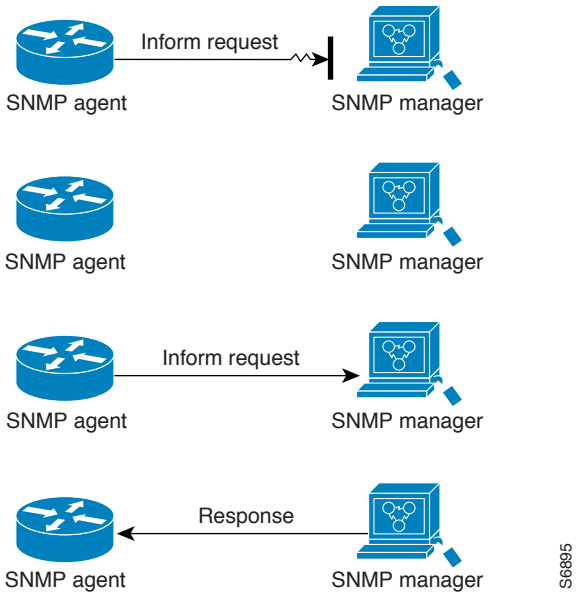
In Figure 17, the agent sends a trap to the manager, but the trap does not reach the manager. Because the agent has no way of knowing that the trap did not reach its destination, the trap is not sent again. The manager never receives the trap.

Figure 17 Trap Unsuccessfully Sent to SNMP Manager



In Figure 18, the agent sends an inform request to the manager, but the inform request does not reach the manager. Because the manager did not receive the inform request, it does not send a response. After a period of time, the agent will resend the inform request. The second time, the manager receives the inform request and replies with a response. In this example, there is more traffic than in Figure 17; however, the notification reaches the SNMP manager.

Figure 18 Inform Request Unsuccessfully Sent to SNMP Manager



MIBs and RFCs

MIB modules typically are defined in RFC documents submitted to the Internet Engineering Task Force (IETF), an international standards body. RFCs are written by individuals or groups for consideration by the Internet Society and the Internet community as a whole, usually with the intention of establishing a recommended Internet standard. Before being given RFC status, recommendations are published as Internet Draft (I-D) documents. RFCs that have become recommended standards are also labeled as standards (STD) documents. You can learn about the standards process and the activities of the IETF at the Internet Society website at http://www.isoc.org. You can read the full text of all RFCs, I-Ds, and STDs referenced in Cisco documentation at the IETF website at http://www.ietf.org.

The Cisco implementation of SNMP uses the definitions of MIB II variables described in RFC 1213 and definitions of SNMP traps described in RFC 1215.

Cisco provides its own private MIB extensions with every system. Cisco enterprise MIBs comply with the guidelines described in the relevant RFCs unless otherwise noted in the documentation. You can find the MIB module definition files and list of which MIBs are supported on each Cisco platform on the Cisco MIB website on Cisco.com.

For a list of new MIB-related functionality, see the "New MIB Features in Cisco IOS Release 12.2" section.

SNMP Versions

Cisco IOS software supports the following versions of SNMP:

- SNMPv1—The Simple Network Management Protocol: A Full Internet Standard, defined in RFC 1157. (RFC 1157 replaces the earlier versions that were published as RFC 1067 and RFC 1098.) Security is based on community strings.
- SNMPv2c—The community-string based Administrative Framework for SNMPv2. SNMPv2c (the "c" stands for "community") is an Experimental Internet Protocol defined in RFC 1901, RFC 1905, and RFC 1906. SNMPv2c is an update of the protocol operations and data types of SNMPv2p (SNMPv2 Classic), and uses the community-based security model of SNMPv1.
- SNMPv3—Version 3 of SNMP. SNMPv3 is an interoperable standards-based protocol defined in RFCs 2273 to 2275. SNMPv3 provides secure access to devices by a combination of authenticating and encrypting packets over the network.

  The security features provided in SNMPv3 are as follows:

  - Message integrity—Ensuring that a packet has not been tampered with in transit.
  - Authentication—Determining that the message is from a valid source.
  - Encryption—Scrambling the contents of a packet prevent it from being learned by an unauthorized source.

Both SNMPv1 and SNMPv2c use a community-based form of security. The community of managers able to access the agent MIB is defined by an IP address Access Control List and password.

SNMPv2c support includes a bulk retrieval mechanism and more detailed error message reporting to management stations. The bulk retrieval mechanism supports the retrieval of tables and large quantities of information, minimizing the number of round-trips required. The SNMPv2C improved error handling support 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. Three kinds of exceptions are also reported: no such object exceptions, no such instance exceptions, and end of MIB view exceptions.

SNMPv3 is a security model.A security model is an authentication strategy that is set up for a user and the group in which the user resides. A security level is the permitted level of security within a security model. A combination of a security model and a security level will determine which security mechanism is employed when handling an SNMP packet. See Table 20 for a list of security levels available in SNMPv3.

Three security models are available: SNMPv1, SNMPv2c, and SNMPv3. Table 20 identifies what the combinations of security models and levels mean.

Table 20 SNMP Security Models and Levels

Model	Level	Authentication	Encryption	What Happens
v1	noAuthNoPriv	Community String	No	Uses a community string match for authentication.
v2c	noAuthNoPriv	Community String	No	Uses a community string match for authentication.
v3	noAuthNoPriv	Username	No	Uses a username match for authentication.
v3	authNoPriv	MD5 or SHA	No	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms.
v3	authPriv	MD5 or SHA	DES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides DES 56-bit encryption in addition to authentication based on the CBC-DES (DES-56) standard.

Note SNMPv2p (SNMPv2 Classic) is not supported in any Cisco IOS releases after 11.2. SNMPv2c replaces the Party-based Administrative and Security Framework of SNMPv2p with a Community-based Administrative Framework. SNMPv2c retained the bulk retrieval and error handling capabilities of SNMPv2p.

You must configure the SNMP agent to use the version of SNMP supported by the management station. An agent can communicate with multiple managers; for this reason, you can configure the Cisco IOS software to support communications with one management station using the SNMPv1 protocol, one using the SNMPv2c protocol, and another using SMNPv3.

The SNMPv3 feature supports RFCs 1901 to 1908, 2104, 2206, 2213, 2214, and 2271 to 2275. For additional information on SNMPv3, refer to RFC 2570, *Introduction to Version 3 of the Internet-standard Network Management Framework* (note that this is not a standards document).

SNMP Configuration Task List

There is no specific command that you use to enable SNMP. The first **snmp-server** command that you enter enables the supported versions of SNMP.

To configure SNMP support, perform the tasks described in the following sections. Each task is labeled as required or optional.

- Creating or Modifying an SNMP View Record (Optional)
- Creating or Modifying Access Control for an SNMP Community (Required)
- Specifying an SNMP-Server Engine Name (ID) (Optional)

- Specifying SNMP-Server Group Names (Optional)
- Configuring SNMP-Server Hosts (Required)
- Configuring SNMP-Server Users (Optional)
- Enabling the SNMP Agent Shutdown Mechanism (Optional)
- Setting the Contact, Location, and Serial Number of the SNMP Agent (Optional)
- Defining the Maximum SNMP Agent Packet Size (Optional)
- Limiting the Number of TFTP Servers Used via SNMP (Optional)
- Monitoring and Troubleshooting SNMP Status (Optional)
- Disabling the SNMP Agent (Optional)
- Configuring SNMP Notifications (Required)
- Configuring the Router as an SNMP Manager (Optional)

Creating or Modifying an SNMP View Record

You can assign views to community strings to limit which MIB objects an SNMP manager can access. You can use a predefined view, or create your own view. If you are using a predefined view or no view at all, skip this task.

To create or modify an SNMP view record, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server view` *view-name oid-tree* `{included | excluded}`	Creates or modifies a view record.

To remove a view record, use the **no snmp-server view** command.

You can enter this command multiple times for the same view record. Later lines take precedence when an object identifier is included in two or more lines.

Creating or Modifying Access Control for an SNMP Community

Use an SNMP community string to define the relationship between the SNMP manager and the agent. The community string acts like a password to regulate access to the agent on the router. Optionally, you can specify one or more of the following 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.

To configure a community string, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server community` *string* `[view` *view-name*`] [ro | rw] [`*number*`]`	Defines the community access string.

You can configure one or more community strings. To remove a specific community string, use the **no snmp-server community** command.

Note The @ symbol is used as a delimiter between the community string and the context in which it is used. For example, specific VLAN information in BRIDGE-MIB may be polled using community@VLAN_ID (for example, public@100) where 100 is the VLAN number. Avoid using the @ symbol as part of the SNMP community string when configuring the **snmp-server community** command.

For an example of configuring a community string, see the "SNMP Configuration Examples" section.

Specifying an SNMP-Server Engine Name (ID)

To specify an identification name (ID) for a local SNMP engine, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server engineID local engineid-string`	Specifies the name of the local SNMP engine (or copy of SNMP).

To specify an ID for a remote SNMP engine, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server engineID remote ip-address [udp-port port-number] engineid-string`	Specifies the name of the remote SNMP engine (or copy of SNMP).

Specifying SNMP-Server Group Names

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

Command	Purpose
`Router(config)# snmp-server group [groupname {v1 | v2c | v3 [auth | noauth | priv]}] [read readview] [write writeview] [notify notifyview] [access access-list]`	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
`Router(config)# snmp-server host host-id [traps | informs][version {1 | 2c | 3 [auth | noauth | priv]} ] community-string [udp-port port-number] [notification-type]`	Specifies whether you want the SNMP notifications sent as traps or informs, the version of SNMP to use, the security level of the notifications (for SNMPv3), and the recipient (host) of the notifications.

Note The @ symbol is used as a delimiter between the community string and the context in which it is used. For example, specific VLAN information in BRIDGE-MIB may be polled using community@VLAN_ID (for example, public@100) where 100 is the VLAN number. Avoid using the @ symbol as part of the SNMP community string when configuring the **snmp-server host** command.

Configuring SNMP-Server Users

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

Command	Purpose
`Router(config)# snmp-server user username groupname [remote ip-address [udp-port port]] {v1 | v2c | v3 [encrypted] [auth {md5 | sha} auth-password ]} [access access-list]`	Configures a new user to an SNMP group.

Enabling the SNMP Agent Shutdown Mechanism

Using SNMP packets, a network management tool can send messages to users on virtual terminals and the console. This facility operates in a similar fashion to the **send** EXEC command; however, the SNMP request that causes the message to be issued to the users also specifies the action to be taken after the message is delivered. One possible action is a shutdown request. After a system is shut down, typically it is reloaded. Because the ability to cause a reload from the network is a powerful feature, it is protected by the **snmp-server system-shutdown** global configuration command. If you do not issue this command, the shutdown mechanism is not enabled. To enable the SNMP agent shutdown mechanism, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server system-shutdown`	Enables system shutdown using the SNMP message reload feature.

Setting the Contact, Location, and Serial Number of the SNMP Agent

You can set the system contact, location, and serial number of the SNMP agent so that these descriptions can be accessed through the configuration file. To do so, use the following commands in global configuration mode, as needed:

Command	Purpose
`Router(config)# snmp-server contact text`	Sets the system contact string.
`Router(config)# snmp-server location text`	Sets the system location string.
`Router(config)# snmp-server chassis-id number`	Sets the system serial number.

Defining the Maximum SNMP Agent Packet Size

You can define the maximum packet size permitted when the SNMP agent is receiving a request or generating a reply. To do so, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server packetsize byte-count`	Establishes the maximum packet size.

Limiting the Number of TFTP Servers Used via SNMP

You can limit the number of TFTP servers used for saving and loading configuration files via SNMP to the servers specified in an access list. To do so, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server tftp-server-list number`	Limits the number of TFTP servers used for configuration file copies via SNMP to the servers in an access list.

Monitoring and Troubleshooting SNMP Status

To monitor and troubleshoot SNMP status and information, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router> show snmp`	Monitors SNMP status.
`Router> show snmp engineID [local | remote]`	Displays information about the local SNMP engine and all remote engines that have been configured on the device.
`Router> show snmp groups`	Displays information about each SNMP group on the network.
`Router> show snmp user`	Displays information about each SNMP username in the SNMP users table.

To monitor SNMP trap activity in real time for the purposes of troubleshooting, use the SNMP **debug** commands, including the **debug snmp packet** EXEC command. For documentation of SNMP **debug** commands, see the *Cisco IOS Debug Command Reference*.

Disabling the SNMP Agent

To disable any version of the SNMP agent, use the following command in global configuration mode:

Command	Purpose
`Router(config)# no snmp-server`	Disables SNMP agent operation.

Configuring SNMP Notifications

To configure the router to send SNMP traps or informs, perform the tasks described in the following sections:

- Configuring the Router to Send SNMP Notifications (Required)
- Changing Notification Operation Values (Optional)
- Controlling Individual RFC 1157 SNMP Traps (Optional)

Note Most Cisco IOS commands use the word "traps" in their command syntax. Unless there is an option within the command to specify either traps or informs, the keyword **traps** should be taken to mean either traps or informs, or both. Use the **snmp-server host** command to specify whether you want SNMP notifications to be sent as traps or informs.
The SNMP Proxy manager must be available and enabled on the device for informs to be used. The SNMP Proxy manager is shipped with PLUS software images only.

Configuring the Router to Send SNMP Notifications

To configure the router to send traps or informs to a host, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# snmp-server engineID remote` *remote-ip-addr remote-engineID*	Specifies the engine ID for the remote host.
Step 2	`Router(config)# snmp-server user` *username groupname* `[remote` *host* `[udp-port` *port*`] {v1 | v2c | v3 [encrypted] [auth {md5 | sha} auth-password]} [access` *access-list*`]`	Configures an SNMP user to be associated with the host created in Step 1. **Note** You cannot configure a remote user for an address without first configuring the engine ID for that remote host . This is a restriction imposed in the design of these commands; if you try to configure the user before the host, you will receive a warning message and the command will not be executed
Step 3	`Router(config)# snmp group` *groupname* `{v1 | v2 | v3 {auth | noauth | priv}} [read` *readview*`] [write` *writeview*`] [notify` *notifyview*`] [access` *access-list*`]`	Configures an SNMP group.

	Command	Purpose
Step 4	`Router(config)# snmp-server host host [traps | informs] [version {1 | 2c | 3 [auth | noauth | priv]}] community-string [notification-type]`	Specifies whether you want the SNMP notifications sent as traps or informs, the version of SNMP to use, the security level of the notifications (for SNMPv3), and the recipient (host) of the notifications.
Step 5	`Router(config)# snmp-server enable traps [notification-type [notification-options]]`	Enables sending of traps or informs, and specifies the type of notifications to be sent. If a *notification-type* is not specified, all supported notification will be enabled on the router. To discover which notifications are available on your router, enter the **snmp-server enable traps ?** command.

The **snmp-server host** command specifies which hosts will receive SNMP notifications, and whether you want the notifications sent as traps or inform requests. The **snmp-server enable traps** command globally enables the production mechanism for the specified notification types (such as Border Gateway Protocol [BGP] traps, config traps, entity traps, Hot Standby Router Protocol [HSRP] traps, and so on).

Changing Notification Operation Values

You can specify a value other than the default for the source interface, message (packet) queue length for each host, or retransmission interval.

To change notification operation values, use the following commands in global configuration mode, as needed:

Command	Purpose
`Router(config)# snmp-server trap-source interface`	Specifies a source interface for trap or inform notifications.
`Router(config)# snmp-server queue-length length`	Establishes the message queue length for each notification.
`Router(config)# snmp-server trap-timeout seconds`	Defines how often to resend notifications on the retransmission queue.

For inform requests, you can configure inform-specific operation values in addition to the operation values mentioned. To change inform operation values, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server informs [retries retries] [timeout seconds] [pending pending]`	Sets the maximum number of times to resend an inform request, the number of seconds to wait for an acknowledgment before resending, and the maximum number of informs waiting for acknowledgments at any one time.

Controlling Individual RFC 1157 SNMP Traps

Starting with Cisco IOS Release 12.1(3)T, you can globally enable or disable authenticationFailure, linkUp, linkDown, warmStart, and coldStart notifications (traps or informs) individually. (These traps constitute the "generic traps" defined in RFC 1157.) To enable any of these notification types, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server enable traps snmp [authentication] [linkup] [linkdown] [warmstart] [coldstart]`	Enables RFC 1157 generic traps. When used without any of the optional keywords, enables authenticationFailure, linkUp, linkDown, warmStart, and coldStart traps. When used with keywords, enables only the trap types specified.

For example, to globally enable only linkUp and linkDown SNMP traps or informs for all interfaces, use the **snmp-server enable traps snmp linkup linkdown** form of this command.

Note that linkUp and linkDown notifications are enabled by default on specific interfaces, but will not be sent unless they are enabled globally. To control (disable or reenable) the sending of linkUp/linkDown notifications for specific interfaces, use the **no snmp trap link-status** command in interface configuration mode.

Configuring the Router as an SNMP Manager

The SNMP manager feature allows a router to act as a network management station. In other words, configuring a router as an SNMP manager allows it to act as an SNMP client. As an SNMP manager, the router can send SNMP requests to agents and receive SNMP responses and notifications from agents. When the SNMP manager process is enabled, the router can query other SNMP agents and process incoming SNMP traps.

Security Considerations

Most network security policies assume that routers will accept SNMP requests, send SNMP responses, and send SNMP notifications.

With the SNMP manager functionality enabled, the router may also send SNMP requests, receive SNMP responses, and receive SNMP notifications. Your security policy implementation may need to be updated prior to enabling this feature.

SNMP requests typically are sent to User Datagram Protocol (UDP) port 161. SNMP responses are typically sent from UDP port 161. SNMP notifications are typically sent to UDP port 162.

SNMP Sessions

Sessions are created when the SNMP manager in the router sends SNMP requests, such as inform requests, to a host, or receives SNMP notifications from a host. One session is created for each destination host. If there is no further communication between the router and host within the session timeout period, the session will be deleted.

The router tracks statistics, such as the average round-trip time required to reach the host, for each session. Using the statistics for a session, the SNMP manager in the router can set reasonable timeout periods for future requests, such as informs, for that host. If the session is deleted, all statistics are lost. If another session with the same host is later created, the request timeout value for replies will return to the default value.

Sessions consume memory. A reasonable session timeout value should be large enough that regularly used sessions are not prematurely deleted, yet small enough such that irregularly used, or one-time sessions, are purged expeditiously.

Enabling the SNMP Manager

To enable the SNMP manager process and set the session timeout value, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# snmp-server manager`	Enables the SNMP manager.
Step 2	`Router(config)# snmp-server manager session-timeout` *seconds*	(Optional) Changes the session timeout value.

Monitoring the SNMP Manager

To monitor the SNMP manager process, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router> show snmp`	Displays global SNMP information.
`Router> show snmp sessions [brief]`	Displays information about current sessions.
`Router> show snmp pending`	Displays information about current pending requests.

SNMP Configuration Examples

The following example enables SNMPv1, SNMPv2c, and SNMPv3. The configuration permits any SNMP manager to access all objects with read-only permissions using the community string named public. This configuration does not cause the router to send any traps.

```
snmp-server community public
```

The following example permits any SNMP to access all objects with read-only permission using the community string named public. The router also will send ISDN traps to the hosts 172.16.1.111 and 172.16.1.33 using SNMPv1 and to the host 172.16.1.27 using SNMPv2c. The community string named public is sent with the traps.

```
snmp-server community public
snmp-server enable traps isdn
snmp-server host 172.16.1.27 version 2c public
snmp-server host 172.16.1.111 version 1 public
snmp-server host 172.16.1.33 public
```

The following example allows read-only access for all objects to members of access list 4 that specify 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 named public.

```
snmp-server community comaccess ro 4
snmp-server enable traps snmp authentication
snmp-server host cisco.com version 2c public
```

The following example sends Entity MIB inform notifications to the host cisco.com. The community string is restricted. The first line enables the router to send Entity MIB notifications in addition to any traps or informs previously enabled. The second line specifies that the notifications should be sent as inform requests, specifies the destination of these informs, and overwrites any previous **snmp-server host** commands for the host cisco.com.

```
snmp-server enable traps entity
snmp-server host informs cisco.com restricted entity
```

The following example sends the SNMP and Cisco environmental monitor enterprise-specific traps to address 172.30.2.160:

```
snmp-server enable traps
snmp-server host 172.30.2.160 public snmp envmon
```

The following example enables the router to send all traps to the host myhost.cisco.com using the community string public:

```
snmp-server enable traps
snmp-server host myhost.cisco.com public
```

The following example will not send traps to any host. The BGP traps are enabled for all hosts, but only the ISDN traps are enabled to be sent to a host.

```
snmp-server enable traps bgp
snmp-server host bob public isdn
```

The following example enables the router to send all inform requests to the host myhost.cisco.com using the community string named public:

```
snmp-server enable traps
snmp-server host myhost.cisco.com informs version 2c public
```

In the following example, the SNMP manager is enabled and the session timeout is set to a larger value than the default:

```
snmp-server manager
snmp-server manager session-timeout 1000
```

New MIB Features in Cisco IOS Release 12.2

This section outlines the new MIBs and MIB enhancements for the current Cisco IOS software release.

Circuit Interface Identification MIB

The Circuit Interface Identification MIB (also known as the Circuit Interface MIB) is a Cisco enterprise MIB used to assist in SNMP monitoring of circuit-based interfaces. The Circuit Interface MIB (CISCO-CIRCUIT-INTERFACE-MIB) provides a MIB object that can be used to identify individual

circuit-based interfaces (for example, interfaces using ATM or Frame Relay). This user-specified identification will then be returned when linkup and linkdown SNMP traps are generated for the interface.

No Cisco IOS software configuration commands are associated with this MIB.

For more information, refer to the CISCO-CIRCUIT-INTERFACE-MIB.my file, available from the Cisco.com MIB website.

Ethernet-like Interfaces MIB

The Ethernet-like Interfaces MIB (ETHERLIKE-MIB) was introduced in Cisco IOS Release 12.1(2)T. The Cisco implementation of the Ethernet-like Interfaces MIB (defined in the ETHERLIKE-MIB.my and CISCO-ETHERLIKE-CAPABILITY.my files on the Cisco MIB website) complies with RFC 2665 (*Definitions of Managed Objects for the Ethernet-like Interface Types*), and Data Over Cable Service Interface Specification (DOCSIS) 1.0 requirements for Cable Modem Termination Systems (CMTSs) and cable modems (CMs). Support for RFC 2665 in the ETHERLIKE-MIB was achieved through the addition of two new objects in the *dot3StatsTable*: *dot3StatsSymbolErrors* and *dot3StatsDuplexStatus*.

No Cisco IOS software configuration commands are associated with this MIB.

Event MIB

The Event MIB was introduced in Cisco IOS Release 12.0(11)S and 12.1(3)T. No Cisco IOS software configuration commands are associated with this MIB. Instead, Event MIB configuration is done with applications external to Cisco IOS software. The Event MIB allows specialized monitoring capabilities that can be configured through a network management system (NMS) application using SNMP Get and Set operations. The Event MIB provides an asynchronous notification mechanism supported by SNMP that can be set to monitor any SNMP MIB object on a Cisco device and perform notification (trap or inform) operations or Set operations when specific conditions occur. Conditions are defined in event values. Event values that have been configured on your system can be displayed using the **show management event** command in privileged EXEC mode. By allowing SNMP notifications to take place only when a specified condition is met, Event MIB support reduces the load on affected devices, substantially improving the scalability of network management solutions.

For further information, see the Event MIB Support feature module document at http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t3/dtevent.htm

Expression MIB Support for Delta, Wildcarding, and Aggregation

Expression MIB adds support of the Delta, Wildcarding, Delta Wildcarding, and Aggregation features in the Distributed Management Expression MIB (EXPRESSION-MIB) to Cisco IOS software for use by SNMP. No Cisco IOS software configuration commands are associated with this MIB. The functionality provided by this MIB is especially useful when used with the Event MIB (described previously).

The Delta function enables the Expression MIB to use Delta values of an object instead of absolute values when evaluating an expression. Delta is obtained by taking the difference in the current value of an object with its previous value. Wildcarding empowers the Expression MIB to evaluate multiple instances of an object. This feature is useful in cases when the expression must be applied to all instances of an object. The user need not individually specify all instances of an object in the Expression but only needs to set the expWildcardedObject in expObjectTable to TRUE for the respective object. Aggregation

is done by using the sum() function in the Expression MIB. The operand to the sum function must be a wildcarded object. The result of the sum() function is the sum of values of all instances of the wildcarded object.

For more information, see the *EXPRESSION-MIB.my* document available from the Cisco.com MIB website.

Interfaces Group MIB Enhancements

The Cisco implementation of the Interfaces Group MIB (IF-MIB) has been enhanced to allow you to enable linkUp and linkDown SNMP traps that are compliant with RFC 2233. The default implementation of linkUp and linkDown traps is defined in CISCO-IF-CAPABILITY.my and OLD-CISCO-INTERFACES- MIB.my. To enable linkUp and linkDown traps that will function for both interfaces and subinterfaces, use the **snmp-server trap link ietf** command in global configuration mode.

The IF-MIB implementation also has been enhanced to allow the consistent identification of interfaces using the Interface Index (ifIndex) value of the IF-MIB.

Interfaces Group MIB Support for ATM Subinterfaces

Introduced in Cisco IOS Release 12.1, the Interfaces Group MIB support for ATM subinterfaces feature provides the implementation of RFC 2233 (MIB-II) for ATM subinterfaces. ATM subinterfaces are visible in the ifTable and accessible to NMS applications. There are two entities in the ifTable corresponding to each subinterface—an atmSubif entity and an aal5 entity. The atmSubif entity corresponds to the ATM layer and the aal5 entity corresponds to the AAL5 layer. The MIB variables are defined in RFC 1695.

MIB Enhancements for Universal Gateways and Access Servers

The following MIB enhancements were designed to monitor modem and line status for network access servers (NASs).

CISCO-AAA-SERVER-MIB

The CISCO-AAA-SERVER-MIB provides statistics reflecting the state of authentication, authorization, and accounting (AAA) server operation within a device and AAA communications with external servers for the Cisco AS5300 and AS5800 series platforms. The Cisco AAA Server MIB provides the following information:

- A table for configuring AAA servers
- Identities of external AAA servers
- Statistics for each AAA function (**show radius statistics** command)
- Status of servers providing AAA functions

ServerStateChange notifications are controlled (enabled or disabled) through use of the **snmp-server enable traps aaa_server** command in global configuration mode. ServerStateChange notifications, when enabled, will be sent when the server moves from an “up” to “dead” state or when a server moves from a “dead” to “up” state.

Statistics for AAA functions can be displayed through use of the **show radius statistics** command in EXEC mode.

The implementation of this MIB is defined in the CISCO-AAA-SERVER-MIB.my and CISCO-AAA-SERVER-CAPABILITY.my files available from the Cisco.com MIB website.

CISCO-AAA-SESSION-MIB

The CISCO-AAA-SESSION-MIB provides the ability to both monitor and terminate authenticated client connections using SNMP for the Cisco AS5300 and AS5800 series platforms. Real-time information can be provided on data such as idle time, allowing configurations that can terminate calls when there are periods of inactivity on a line. Data provided by this MIB is directly related to the accounting information reported by AAA to RADIUS or TACACS servers. You can verify SNMP queried values through use of the **show accounting** and **show caller timeouts** commands in EXEC mode.

To enable the ability to terminate connections, you must configure the device through use of the **aaa session-mib** {**disconnect**} command in global configuration mode. When this command is found in a system configuration, SNMP managers have the ability to disconnect all lines that have AAA accounting records associated to them using the Disconnect object. (AAA must already be configured with accounting enabled for this feature to function.) For more information, see the Release 12.2 *Cisco IOS Security Configuration Guide*.

CISCO-CALL-TRACKER-MIB, CISCO-CALL-TRACKER-MODEM-MIB, and CISCO-CALL-TRACKER-TCP-MIB

The CISCO-CALL-TRACKER-MIB, the CISCO-CALL-TRACKER-MODEM-MIB, and the CISCO-CALL-TRACKER-TCP-MIB provide the ability to capture detailed data on the progress and status of calls, from the time the NAS receives a setup request or allocates a channel, to the time a call is rejected or terminated. This data is maintained within the Call Tracker database tables, which are accessible through SNMP, command-line interface (CLI), or SYSLOG.

Call Tracker SNMP notifications are controlled through use of the **snmp-server enable traps snmp calltracker** command in global configuration mode. CallSetup notifications are generated at the start of each call, when an entry is created in the active table (cctActiveTable), and CallTerminate notifications are generated at the end of each call, when an entry is created in the history table (cctHistoryTable).

The Call Tracker feature is supported on the Cisco AS5300 and the Cisco AS5800 series platforms. For more information on this feature, see the *Call Tracker plus ISDN and AAA Enhancements for the Cisco AS5300 and Cisco AS5800* document available from Cisco.com.

CISCO-ISDN-MIB

The CISCO-ISDN-MIB supplies ISDN PRI channel-not-available traps that can be generated when a requested DS 0 channel is not available, or when no modem is available to take the incoming call. ISDN PRI channel-not-available notifications are controlled (enabled or disabled) through use of the **no snmp-server enable traps isdn** [**chan-not-avail**] command in global configuration mode. These notifications are disabled by default and are available only for ISDN PRI interfaces on the Cisco AS5300, Cisco AS5400, and Cisco AS5800 universal access servers.

CISCO-MODEM-MGMT-MIB

The CISCO-MODEM-MGMT-MIB supplies modem health traps that can be generated when a modem port is bad, disabled, reflashed, or shut down, or when there is a request to busyout the modem. Modem health notifications are controlled (enabled or disabled) through use of the **no snmp-server enable traps modem-health** command in global configuration mode. Modem health traps are disabled by default and are supported on the Cisco AS5300, Cisco AS5400, and Cisco AS5800 universal access servers.

CISCO-POP-MGMT-MIB

The CISCO-POP-MGMT-MIB supplies the DS 0 busyout notification. DS 0 busyout traps or informs can be generated when there is a request to busyout a DS 0, when there is a request to take a DS 0 out of busyout mode, or when busyout completes and the DS 0 is out of service. DS 0 busyout traps are controlled (enabled or disabled) through use of the **no snmp-server enable traps pop** command in global configuration mode. Busyout is enabled on a device using the **isdn snmp busyout b-channel** command. DS 0 busyout notifications are disabled by default and are supported on Cisco AS5300, Cisco AS5400, and Cisco AS5800 universal access servers.

DS 1 loopback traps can be generated when a DS 1 line goes into loopback mode. DS 1 loopback traps are controlled (enabled or disabled) through use of the **no snmp-server enable traps ds1-loopback** command in global configuration mode. DS 1 loopback traps are disabled by default and are supported only on the Cisco AS5300 and Cisco AS5400 universal access servers.

RFC1406-MIB

The RCF1406-MIB supplies dsx1LineStatus and dsx1LineIndex objects.

MSDP MIB

The Multicast Source Discovery Protocol (MSDP) MIB feature adds support in Cisco IOS software for the MSDP MIB. This MIB describes objects used for managing MSDP operations using SNMP. MSDP MIB notifications are controlled (enabled or disabled) through use of the **no snmp-server enable traps msdp** command in global configuration mode. There are two MSDP MIB notification-types: msdpEstablished (1) and msdpBackwardTransition (2). The msdpEstablish notifications are sent when the MSDP finite state machine (FSM) enters the ESTABLISHED state. The msdpBackwardTransition notifications are sent generated when the MSDP FSM moves from a higher numbered state to a lower numbered state. For more information on the Cisco implementation of the MSDP MIB, refer to the *MSDP-MIB.my* document available from Cisco.com. The Cisco implementation of the MSDP MIB has the following restrictions in Cisco IOS Release 12.2:

- All MSDP MIB objects are implemented as read-only.
- The Requests table is not supported in the Cisco implementation of the MSDP MIB.
- The msdpEstablished notification is not supported in the Cisco implementation of the MSDP MIB.

NTP MIB

The Network Time Protocol (NTP) is used to synchronize timekeeping among a set of distributed time servers and clients. The Cisco NTP MIB enables users to remotely monitor an NTP server using SNMP, provided the MIB itself is implemented on that server. Use of the NTP MIB to monitor the NTP status of routing devices is accomplished using software on an NMS. No new or modified Cisco IOS software commands are associated with this feature.

The Cisco implementation of the NTP MIB is based on NTP version 3 (RFC-1305). The MIB objects are all read-only. SNMP requests are processed by reading the corresponding variables from the NTP subsystem and returning them in the response. The NTP MIB defines a set of NTP server system objects, including an NTP server peers table and an NTP server filter register table. For more information on the Cisco implementation of the NTP MIB, refer to the MIB document itself (*CISCO-NTP-MIB.my*, available from Cisco.com).

Response Time Monitor MIB

The CISCO-RTTMON-MIB is used for network monitoring and management using the Cisco Service Assurance Agent (SA Agent). For information about the enhancements to this MIB, see the "Network Monitoring Using Cisco Service Assurance Agent" chapter in this document.