Simple Network Management Protocol

Simple Network Management Protocol (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 that is used for monitoring and managing devices in a network.

This module discusses how to enable an SNMP agent on a Cisco device and how to control the sending of SNMP notifications from the agent. For information about using SNMP management systems, see the appropriate documentation for your network management system (NMS) application.

For a complete description of the device monitoring commands mentioned in this document, see the Cisco Network Management Command Reference. To locate documentation of other commands that appear in this document, use the Cisco IOS Master Command List or search online.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for SNMP

Not all Cisco platforms are supported on the features described in this module. Use Cisco Feature Navigator to find information about platform support and Cisco software image support.

Information About Configuring SNMP Support

Components of SNMP

The Simple Network Management Protocol (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 monitoring and managing devices in a network.

The SNMP framework has the following components, which are described in the following sections:

SNMP Operations

SNMP applications perform the following operations to retrieve data, modify SNMP object variables, and send notifications:

SNMP Get

The Simple Network Management Protocol (SNMP) GET operation is performed by an Network Management Server (NMS) to retrieve SNMP object variables. There are three types of GET operations:

- GET—Retrieves the exact object instance from the SNMP agent.
- GETNEXT—Retrieves the next object variable, which is a lexicographical successor to the specified variable.
- GETBULK—Retrieves a large amount of object variable data, without the need for repeated GETNEXT operations.

SNMP SET

The Simple Network Management Protocol (SNMP) SET operation is performed by a Network Management Server (NMS) to modify the value of an object variable.

SNMP Notifications

A key feature of SNMP is its capability to generate unsolicited notifications from an SNMP agent.

Traps and Informs

Unsolicited (asynchronous) notifications can be generated as traps or inform requests (informs). Traps are messages alerting the SNMP manager to a condition on the network. 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 neighboring device, or other significant events.

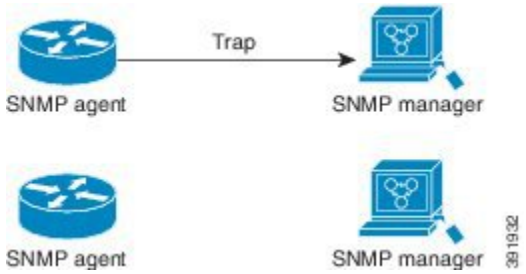
Traps are less reliable than informs because the receiver does not send an acknowledgment when it receives a trap. The sender does not know if the trap was received. An SNMP manager that receives an inform, acknowledges the message with an SNMP response protocol data unit (PDU). If the sender never receives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

Traps are often preferred even though they are less reliable because informs consume more resources in the device and in the network. Unlike a trap, which is discarded as soon as it is sent, an inform must be held in memory until a response is received or the request times out. Also, traps are sent only once, whereas an inform may be resent several times. The retries increase traffic and contribute to higher overhead on the network. Use of traps and informs requires a trade-off between reliability and resources. If it is important that the SNMP manager receives every notification, use informs, but if traffic volume or memory usage are concerns and receipt of every notification is not required, use traps.

The following figures illustrate the differences between traps and informs.

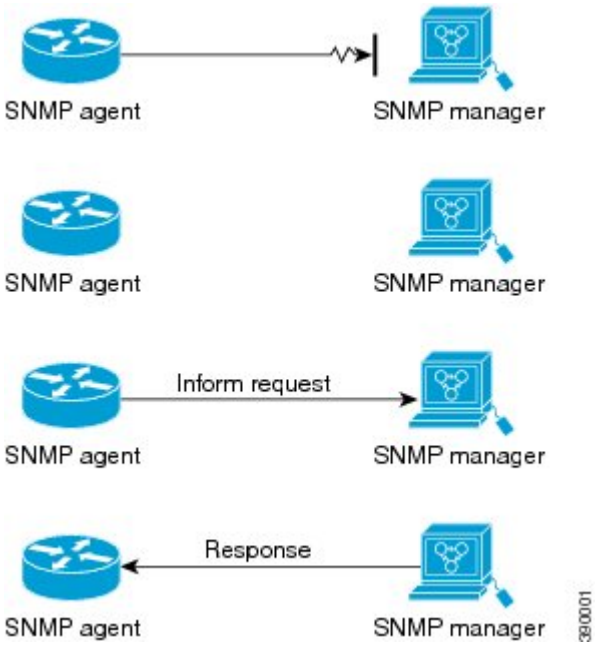
The figure below shows that an agent successfully sends a trap to an SNMP manager. Although the manager receives the trap, it does not send an acknowledgment. The agent has no way of knowing that the trap reached its destination.

Figure 1: Trap Successfully Sent to SNMP Manager



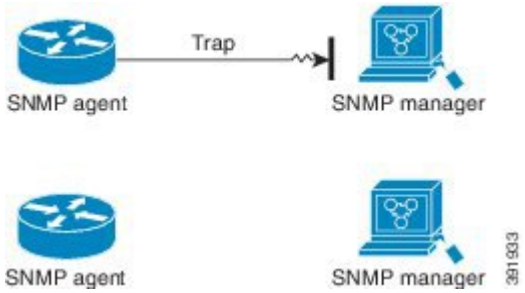
In the figure below, the agent successfully sends an inform to the manager. When the manager receives the inform, a response is sent to the agent and the agent knows that the inform reached its destination. Notice that in this example, the traffic generated is twice as much as in the interaction shown in the table above.

Figure 2: Inform Request Successfully Sent to SNMP Manager



The figure below shows an agent sending a trap to a manager that the manager does not receive. The agent has no way of knowing that the trap did not reach its destination. The manager never receives the trap because traps are not resent.

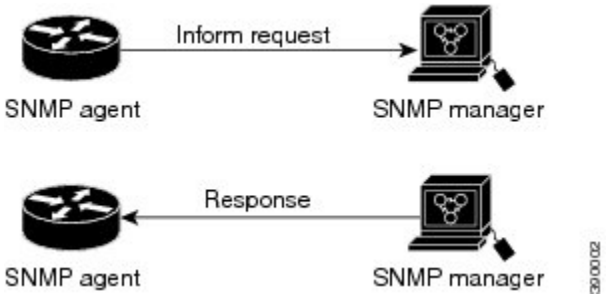
Figure 3: Trap Unsuccessfully Sent to SNMP Manager



The figure below shows an agent sending an inform to a manager that does not reach the manager. Because the manager did not receive the inform, it does not send a response. After a period of time, the agent resends the inform. The manager receives the inform from the second transmission and replies. In this example, more

traffic is generated than in the scenario shown in the table above but the notification reaches the SNMP manager.

Figure 4: Inform Unsuccessfully Sent to SNMP Manager



Versions of SNMP

The Cisco IOS software supports the following versions of SNMP:

- SNMPv1—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" is 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 3413 to 3415. SNMPv3 provides secure access to devices by 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 to prevent it from being learned by an unauthorized source.

Both SNMPv1 and SNMPv2c use a community-based form of security. The community of SNMP managers able to access the agent MIB is defined by a community string.

SNMPv2c support includes a bulk retrieval mechanism and 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 types of errors; these conditions are reported through a single error code in SNMPv1. The following three types of exceptions are also reported: no such object, no such instance, and end of MIB view.

SNMPv3 is a security model in which an authentication strategy 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 determines which security mechanism is employed when handling an SNMP packet.

Three security models are available: SNMPv1, SNMPv2c, and SNMPv3. The table below lists the combinations of security models and levels and their meanings.

Table 1: 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	Message Digest 5 (MD5) or Secure Hash Algorithm (SHA)	No	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms.
v3	authPriv	MD5 or SHA	Data Encryption Standard (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 Cisco IOS Release 11.2 and later releases. 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 an SNMP agent to use the version of SNMP supported by the management station. An agent can communicate with multiple managers. 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 SNMPv3.

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

How to Configure SNMP Support

There is no specific command to enable SNMP. The first **snmp-server** command that you enter enables supported versions of SNMP. All other configurations are optional.

Configuring System Information

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. Although the configuration steps described in this section are optional, configuring the basic information is recommended because it may be useful when troubleshooting your configuration. In addition, the first **snmp-server** command that you issue enables SNMP on the device.

Perform this task as needed.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server contact** *text*
4. **snmp-server location** *text*
5. **snmp-server chassis-id** *number*
6. end
7. **show snmp contact**
8. **show snmp location**
9. **show snmp chassis**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**snmp-server contact** *text* **Example:** `Device(config)# snmp-server contact NameOne`	Sets the system contact string.
Step 4	**snmp-server location** *text* **Example:** `Device(config)# snmp-server location LocationOne`	Sets the system location string.
Step 5	**snmp-server chassis-id** *number* **Example:** `Device(config)# snmp-server chassis-id 015A619T`	Sets the system serial number.
Step 6	end **Example:** `Device(config)# end`	Exits global configuration mode.
Step 7	**show snmp contact** **Example:** `Device# show snmp contact`	(Optional) Displays the contact strings configured for the system.
Step 8	**show snmp location** **Example:** `Device# show snmp location`	(Optional) Displays the location string configured for the system.
Step 9	**show snmp chassis** **Example:** `Device# show snmp chassis`	(Optional) Displays the system serial number.

Enabling the SNMP Agent Shutdown Mechanism

Using SNMP packets, a network management tool can send messages to users on virtual terminals and on 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.

Perform this task to enable the SNMP agent shutdown mechanism.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server system-shutdown**
4. end

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server system-shutdown** **Example:** `Device(config)# snmp-server system-shutdown`	Enables system shutdown using the SNMP message reload feature.
Step 4	end **Example:** `Device(config)# end`	Exits global configuration mode.

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.

Perform this task to set the maximum permitted packet size.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server packetsize** *byte-count*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server packetsize** *byte-count* **Example:** `Device(config)# snmp-server packetsize 512`	Establishes the maximum packet size.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

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 by using an access list. Limiting the use of TFTP servers in this way conserves system resources and centralizes the operation for manageability.

Perform this task to limit the number of TFTP servers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server tftp-server-list** *number*
4. end

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server tftp-server-list** *number* **Example:** `Device(config)# snmp-server tftp-server-list 12`	Limits the number of TFTP servers used for configuration file copies via SNMP to the servers in an access list.
Step 4	end **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

An alternative to using the ifAlias value for the identification of interfaces across reboots is to use the cciDescr object in the Cisco Circuit Interface MIB (CISCO-CIRCUIT-INTERFACE-MIB.my). This MIB object can be used only for circuit-based interfaces such as ATM or Frame Relay interfaces. Cisco IOS feature FTS-731 introduced the Circuit Interface Identification Persistence for the Simple Network Management Protocol (SNMP), which maintains the user-defined name of the circuit (defined in the cciDescr object) across reboots and allows consistent identification of circuit-based interfaces.

Configuring SNMP Versions 1 and 2

When you configure SNMP versions 1 and 2, you can optionally create or modify views for community strings to limit which MIB objects an SNMP manager can access.

Perform the following tasks when configuring SNMP version 1 or version 2.

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.

Perform this task to create or modify an SNMP view record.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server view** *view-name oid-tree* {**included** | **excluded**}
4. **no snmp-server view** *view-name oid-tree* {**included** | **excluded**}
5. **end**
6. show snmp view

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server view** *view-name oid-tree* {**included** | **excluded**} **Example:** `Device(config)# snmp-server view mib2 mib-2 included`	Creates a view record. • In this example, the mib2 view that includes all objects in the MIB-II subtree is created. **Note** You can use this command multiple times to create the same view record. If a view record for the same OID value is created multiple times, the latest entry of the object identifier takes precedence.

	Command or Action	Purpose
Step 4	**no snmp-server view** *view-name oid-tree* {**included** | **excluded**} **Example:** `Device(config)# no snmp-server view mib2 mib-2 included`	Removes a server view.
Step 5	**end** **Example:** `Device(config)# end`	Exits global configuration mode.
Step 6	show snmp view **Example:** `Device# show snmp view`	(Optional) Displays a view of the MIBs associated with SNMP.

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 device. 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.

Perform this task to create or modify a community string.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [**ipv6** *nacl*] [*access-list-number*]
4. **no snmp-server community** *string*
5. **end**
6. **show snmp community**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [**ipv6** *nacl*] [*access-list-number*] **Example:** `Device(config)# snmp-server community comaccess ro 4`	Defines the community access string. • You can configure one or more community strings.
Step 4	**no snmp-server community** *string* **Example:** `Device(config)# no snmp-server community comaccess`	Removes the community string from the configuration.
Step 5	**end** **Example:** `Device(config)# end`	Exits global configuration mode.
Step 6	**show snmp community** **Example:** `Device# show snmp community`	(Optional) Displays the community access strings configured for the system.

Configuring a Recipient of an SNMP Trap Operation

To enable multiple hosts, you must issue a separate **snmp-server host** command for each host. You can specify multiple notification types in the command for each host.

When multiple **snmp-server host** commands are given for the same host and kind of notification (trap or inform), each succeeding command overwrites the previous command. Only the last **snmp-server host** command will be in effect. For example, if you enter an **snmp-server host inform** command for a host and

then enter another **snmp-server host inform** command for the same host, the second command will replace the first.

The **snmp-server host** command is used in conjunction with the **snmp-server enable** command. Use the **snmp-server enable** command to specify which SNMP notifications are sent globally. For a host to receive most notifications, at least one **snmp-server enable** command and the **snmp-server host** command for that host must be enabled.

Some notification types cannot be controlled with the **snmp-server enable** command. For example, some notification types are always enabled and others are enabled by a different command. For example, the linkUpDown notifications are controlled by the **snmp trap link-status** interface configuration command. These notification types do not require an **snmp-server enable** command.

A *notification-type* option's availability depends on the device type and Cisco IOS software features supported on the device. For example, the Cisco IOS software does not support the envmon notification type. To see what notification types are available on your system, use the command help (?) at the end of the **snmp-server host** command.

Perform this task to configure the recipient of an SNMP trap operation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server host** *host-id* [**traps** | **informs**][**version** {**1**| **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port-number*] [*notification-type*]
4. **end**
5. **show snmp host**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server host** *host-id* [**traps**	**informs**][**version** {**1**| **2c**	**3** [**auth**	**noauth**	**priv**]}] *community-string* [**udp-port** *port-number*] [*notification-type*] **Example:** `Device(config)# snmp-server host 172.16.1.27 version 2c public`	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.

	Command or Action	Purpose
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode.
Step 5	**show snmp host** **Example:** `Device# show snmp host`	(Optional) Displays the SNMP notifications sent as traps, the version of SNMP, and the host IP address of the notifications.

Disabling the SNMP Agent

Perform this task to disable any version of an SNMP agent.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no snmp-server**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no snmp-server** **Example:** `Device(config)# no snmp-server`	Disables SNMP agent operation.

	Command or Action	Purpose
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode.

Configuration Examples for SNMP Support

Example: Configuring SNMPv1 Support

The following example shows how to enable SNMPv1. 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 traps.

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

The following example shows how to permit SNMP access to all objects with read-only permission using the community string named public. The router also will send BGP traps to the hosts 172.16.1.111 and 172.16.1.33 using SNMPv1. The community string named public is sent with the traps.

```
Device(config)# snmp-server community public
Device(config)# snmp-server enable traps bgp
Device(config)# snmp-server host 172.16.1.111 version 1 public
Device(config)# snmp-server host 172.16.1.33 public
```

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

```
Device(config)# snmp-server enable traps
Device(config)# snmp-server host 172.30.2.160 public snmp envmon
```

The following example shows how to enable the router to send all traps to the host example.com using the community string public:

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

The following example shows a configuration in which no traps are sent to a host. The BGP traps are enabled for all hosts, but only the OSPF traps are enabled to be sent to a host.

```
Device(config)# snmp-server enable traps bgp
Device(config)# snmp-server host host1 public ospf
```

The following example shows how to enable a router to send all informs to the host example.com using the community string named public:

```
Device(config)# snmp-server enable traps
Device(config)# snmp-server host example.com informs version 2c public
```

The following example shows how to enable the SNMP manager and set the session timeout to a value greater than the default:

```
Device(config)# snmp-server manager
Device(config)# snmp-server manager session-timeout 1000
```

The following example shows how to enable the SNMP manager to access all objects with read-only permissions. The user is specified as *abcd* and the authentication password is *abcdpasswd*. To obtain the automatically generated default local engine ID, use the **show snmp engineID** command.

```
Device(config)# snmp-server view readview internet included
Device(config)# snmp-server view readview iso included
Device(config)# snmp-server group group1 v3 noauth read readview
Device(config)# snmp-server user abcd group1 v3 auth md5 abcdpasswd
```

Example: Show SNMP View

The following example shows the SNMP view for the system OID tree:

```
Device> enable
Device# configure terminal
Device(config)# snmp-server view test system included
Device(config)# end
Device# show snmp view

test system - included nonvolatile active
cac_view pimMIB - included read-only active
cac_view msdpMIB - included read-only active
cac_view interfaces - included read-only active
cac_view ip - included read-only active
cac_view ospf - included read-only active
.
.
.
v1default iso - included permanent active
v1default internet - included permanent active
v1default snmpUsmMIB - excluded permanent active
v1default snmpVacmMIB - excluded permanent active
v1default snmpCommunityMIB - excluded permanent active
v1default ciscoIpTapMIB - excluded permanent active
v1default ciscoMgmt.395 - excluded permanent active
v1default ciscoTap2MIB - excluded permanent active
.
.
.
```

Example Configuring SNMP Community Access Strings

The following example shows the community access strings configured to enable access to the SNMP manager:

```
Device> enable
Device# configure terminal
Device(config)# snmp-server community public ro
Device(config)# snmp-server community private rw
Device(config)# end
Device# show snmp community

Community name: private
```

```
Community Index: private
Community SecurityName: private
storage-type: nonvolatile active
Community name: public
Community Index: public
Community SecurityName: public
storage-type: nonvolatile active
```

Example Configuring Host Information

The following example shows the host information configured for SNMP notifications:

```
Device> enable
Device# configure terminal
Device(config)# snmp-server host 10.2.28.1 inform version 2c public
Device(config)# end
Device# show snmp host

Notification host: 10.2.28.1 udp-port: 162   type: inform
user: public    security model: v2c
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
SNMP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS SNMP Command Reference
Cisco implementation of RFC 1724, RIP Version 2 MIB Extensions	RIPv2 Monitoring with SNMP Using the RFC 1724 MIB Extensions feature module
DSP Operational State Notifications for notifications to be generated when a digital signaling processor (DSP) is used	DSP Operational State Notifications feature module

Standards and RFCs

Standard/RFC	Title
CBC-DES (DES-56) standard	*Symmetric Encryption Protocol*
STD: 58	*Structure of Management Information Version 2 (SMIv2)*
RFC 1067	*A Simple Network Management Protocol*

Standard/RFC	Title
RFC 1091	*Telnet terminal-type option*
RFC 1098	*Simple Network Management Protocol (SNMP)*
RFC 1157	*Simple Network Management Protocol (SNMP)*
RFC 1213	*Management Information Base for Network Management of TCP/IP-based internets:MIB-II*
RFC 1215	*Convention for defining traps for use with the SNMP*
RFC 1901	*Introduction to Community-based SNMPv2*
RFC 1905	*Common Management Information Services and Protocol over TCP/IP (CMOT)*
RFC 1906	*Telnet X Display Location Option*
RFC 1908	*Simple Network Management Protocol (SNMP)*
RFC 2104	*HMAC: Keyed-Hashing for Message Authentication*
RFC 2206	*RSVP Management Information Base using SMIv2*
RFC 2213	*Integrated Services Management Information Base using SMIv2*
RFC 2214	*Integrated Services Management Information Base Guaranteed Service Extensions using SMIv2*
RFC 2271	*An Architecture for Describing SNMP Management Frameworks*
RFC 2570	*Introduction to Version 3 of the Internet-standard Network Management Framework*
RFC 2578	*Structure of Management Information Version 2 (SMIv2)*
RFC 2579	*Textual Conventions for SMIv2*
RFC 2580	*Conformance Statements for SMIv2*
RFC 2981	*Event MIB*
RFC 2982	*Distributed Management Expression MIB*
RFC 3413	*SNMPv3 Applications*
RFC 3415	*View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*

Standard/RFC	Title
RFC 3418	*Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)*

MIBs

MIB	MIBs Link
• Circuit Interface Identification MIB • Cisco SNMPv2 • Ethernet-like Interfaces MIB • Event MIB • Expression MIB Support for Delta, Wildcarding, and Aggregation • Interfaces Group MIB (IF-MIB) • Interfaces Group MIB Enhancements • MIB Enhancements for Universal Gateways and Access Servers • MSDP MIB • NTP MIB • Response Time Monitor MIB • Virtual Switch MIB	To locate and download MIBs for selected platforms, releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Simple Network Management Protocol

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to . An account on Cisco.com is not required.

Table 2: Feature Information for Simple Network Management Protocol

Feature Name	Releases	Feature Information
SNMP (Simple Network Management Protocol)	Cisco IOS XE Release 3.3SE	The Simple Network Management Protocol (SNMP) feature provides an application-layer protocol that facilitates the exchange of management information between network devices. SNMP is part of the TCP/IP protocol suite. SNMP enables network administrators to manage network performance, find and solve network problems, and plan for network growth. In Cisco IOS XE Release 3.3SE, support was added for the Cisco Catalyst 3650 Series Switches and Cisco Catalyst 3850 Series Switches.