Configure SNMP

This chapter explains Simple Network Management Protocol (SNMP) as implemented by NCS 4000.

Prerequisites for Implementing SNMP

You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Restrictions for SNMP Use on Cisco IOS XR Software

SNMP outputs are only 32-bits wide and therefore cannot display any information greater than 2^{32}. 2^{32} is equal to 4.29 Gigabits. Note that a 10 Gigabit interface is greater than this and so if you are trying to display speed information regarding the interface, you might see concatenated results.

Information About Implementing SNMP

To implement SNMP, you need to understand the concepts described in this section.

SNMP Functional Overview

The SNMP framework consists of three parts:

- SNMP manager
- SNMP agent
- Management Information Base (MIB)

SNMP Manager

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.

SNMP Agent

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 router. To enable the SNMP agent, you must define the relationship between the manager and the agent.

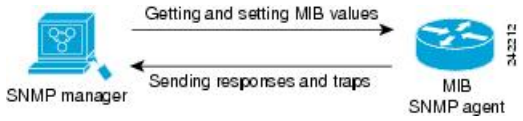
MIB

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. 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.

The figure below, 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) to the manager to notify the manager of network conditions.

Figure 1: Communication Between an SNMP Agent and Manager



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. On Cisco IOS XR software, unsolicited (asynchronous) notifications can be generated only as *traps*. Traps are messages alerting the SNMP manager to a condition on the network. Notifications can indicate improper user authentication, restarts, the closing of a connection, loss of connection to a neighbor router, or other significant events.

Note Inform requests (inform operations) are not supported in Cisco IOS XR software.

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, and 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.

Figure 2: Trap Received by the SNMP Manager

In this illustration, the agent router 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.

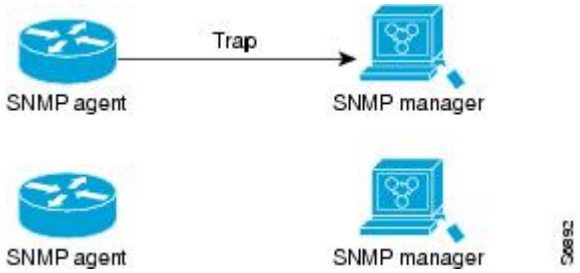
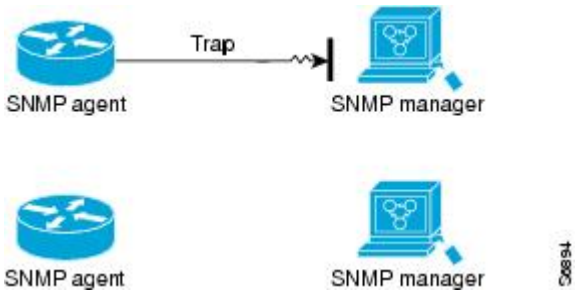


Figure 3: Trap Not Received by the SNMP Manager

In this illustration, 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.



SNMP Versions

Cisco IOS XR software supports the following versions of SNMP:

- Simple Network Management Protocol Version 1 (SNMPv1)
- Simple Network Management Protocol Version 2c (SNMPv2c)
- Simple Network Management Protocol Version 3 (SNMPv3)

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 an SNMP packet is handled. See Security Models and Levels for SNMPv1, v2, v3, on page 5 for a list of security levels available in SNMPv3. The SNMPv3 feature supports RFCs 3411 to 3418.

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-XR software to support communications with one management station using the SNMPv1 protocol, one using the SNMPv2c protocol, and another using SMNPv3.

Comparison of SNMPv1, v2c, and v3

SNMP v1, v2c, and v3 all support the following operations:

- get-request—Retrieves a value from a specific variable.
- get-next-request—Retrieves the value following the named variable; this operation is often used to retrieve variables from within a table. With this operation, an SNMP manager does not need to know the exact variable name. The SNMP manager searches sequentially to find the needed variable from within the MIB.
- get-response—Operation that replies to a get-request, get-next-request, and set-request sent by an NMS.
- set-request—Operation that stores a value in a specific variable.
- trap—Unsolicited message sent by an SNMP agent to an SNMP manager when some event has occurred.

Table 1: SNMPv1, v2c, and v3 Feature Support

Feature	SNMP v1	SNMP v2c	SNMP v3
Get-Bulk Operation	No	Yes	Yes
Inform Operation	No	Yes (No on the Cisco IOS XR software)	Yes (No on the Cisco IOS XR software)
64 Bit Counter	No	Yes	Yes
Textual Conventions	No	Yes	Yes
Authentication	No	No	Yes
Privacy (Encryption)	No	No	Yes

Feature	SNMP v1	SNMP v2c	SNMP v3
Authorization and Access Controls (Views)	No	No	Yes

Security Models and Levels for SNMPv1, v2, v3

The security level determines if an SNMP message needs to be protected from disclosure and if the message needs to be authenticated. The various security levels that exist within a security model are as follows:

- noAuthNoPriv—Security level that does not provide authentication or encryption.
- authNoPriv—Security level that provides authentication but does not provide encryption.
- authPriv—Security level that provides both authentication and encryption.

Three security models are available: SNMPv1, SNMPv2c, and SNMPv3. The security model combined with the security level determine the security mechanism applied when the SNMP message is processed.

Table 2: 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	HMAC-MD5 or HMAC-SHA	No	Provides authentication based on the HMAC[1]-MD5[2] algorithm or the HMAC-SHA[3].
v3	authPriv	HMAC-MD5 or HMAC-SHA	DES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides DES[4] 56-bit encryption in addition to authentication based on the CBC[5] DES (DES-56) standard.
v3	authPriv	HMAC-MD5 or HMAC-SHA	3DES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides 168-bit 3DES[6] level of encryption.
v3	authPriv	HMAC-MD5 or HMAC-SHA	AES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides 128-bit AES[7] level of encryption.

1 Hash-Based Message Authentication Code
2 Message Digest 5
3 Secure Hash Algorithm
4 Data Encryption Standard
5 Cipher Block Chaining
6 Triple Data Encryption Standard

[7] Advanced Encryption Standard

SNMPv3 Benefits

SNMPv3 provides secure access to devices by providing authentication, encryption and access control. These added security benefits secure SNMP against the following security threats:

- Masquerade—The threat that an SNMP user may assume the identity of another SNMP user to perform management operations for which that SNMP user does not have authorization.
- Message stream modification—The threat that messages may be maliciously reordered, delayed, or replayed (to an extent that is greater than can occur through the natural operation of a subnetwork service) to cause SNMP to perform unauthorized management operations.
- Disclosure—The threat that exchanges between SNMP engines could be eavesdropped. Protecting against this threat may be required as a matter of local policy.

In addition, SNMPv3 provides access control over protocol operations on SNMP managed objects.

SNMPv3 Costs

SNMPv3 authentication and encryption contribute to a slight increase in the response time when SNMP operations on MIB objects are performed. This cost is far outweighed by the security advantages provided by SNMPv3.

Table 3: Order of Response Times from Least to Greatest

Security Model	Security Level
SNMPv2c	noAuthNoPriv
SNMPv3	noAuthNoPriv
SNMPv3	authNoPriv
SNMPv3	authPriv

User-Based Security Model

SNMPv3 User-Based Security Model (USM) refers to SNMP message-level security and offers the following services:

- Message integrity—Ensures that messages have not been altered or destroyed in an unauthorized manner and that data sequences have not been altered to an extent greater than can occur nonmaliciously.
- Message origin authentication—Ensures that the claimed identity of the user on whose behalf received data was originated is confirmed.
- Message confidentiality—Ensures that information is not made available or disclosed to unauthorized individuals, entities, or processes.

SNMPv3 authorizes management operations only by configured users and encrypts SNMP messages.

USM uses two authentication protocols:

- HMAC-MD5-96 authentication protocol
- HMAC-SHA-96 authentication protocol

USM uses Cipher Block Chaining (CBC)-DES (DES-56) as the privacy protocol for message encryption.

View-Based Access Control Model

The View-Based Access Control Model (VACM) enables SNMP users to control access to SNMP managed objects by supplying read, write, or notify access to SNMP objects. It prevents access to objects restricted by views. These access policies can be set when user groups are configured with the **snmp-server group** command.

MIB Views

For security reasons, it is often valuable to be able to restrict the access rights of some groups to only a subset of the management information within the management domain. To provide this capability, access to a management object is controlled through MIB views, which contain the set of managed object types (and, optionally, the specific instances of object types) that can be viewed.

Access Policy

Access policy determines the access rights of a group. The three types of access rights are as follows:

- read-view access—The set of object instances authorized for the group when objects are read.
- write-view access—The set of object instances authorized for the group when objects are written.
- notify-view access—The set of object instances authorized for the group when objects are sent in a notification.

IP Precedence and DSCP Support for SNMP

SNMP IP Precedence and differentiated services code point (DSCP) support delivers QoS specifically for SNMP traffic. You can change the priority setting so that SNMP traffic generated in a router is assigned a specific QoS class. The IP Precedence or IP DSCP code point value is used to determine how packets are handled in weighted random early detection (WRED).

After the IP Precedence or DSCP is set for the SNMP traffic generated in a router, different QoS classes cannot be assigned to different types of SNMP traffic in that router.

The IP Precedence value is the first three bits in the type of service (ToS) byte of an IP header. The IP DSCP code point value is the first six bits of the differentiate services (DiffServ Field) byte. You can configure up to eight different IP Precedence markings or 64 different IP DSCP markings.

How to Implement SNMP on Cisco IOS XR Software

This section describes how to implement SNMP.

The **snmp-server** commands enable SNMP on Management Ethernet interfaces by default.

Configuring SNMPv3

This task explains how to configure SNMPv3 for network management and monitoring.

Note No specific command enables SNMPv3; the first **snmp-server** global configuration command (config), that you issue enables SNMPv3. Therefore, the sequence in which you issue the **snmp-server** commands for this task does not matter.

Procedure

Step 1 **configure**

Step 2 **snmp-server view** *view-name oid-tree* {**included** | **excluded**}

Example:

```
RP/0/RP0:hostname(config)# snmp-server view
view_name 1.3.6.1.2.1.1.5 included
```

Creates or modifies a view record.

Step 3 **snmp-server group** *name* {**v1** | **v2c** | **v3** {**ipv4** | **ipv6** | **context**}} [**read** *view*] [**write** *view*] [**notify** *view*] [*access-list-name*]

Example:

```
RP/0/RP0:hostname(config)# snmp-server group
group_name v3 noauth read view_name1 write view_name2
```

Configures a new SNMP group or a table that maps SNMP users to SNMP views.

Step 4 **snmp-server user** *username groupname* {**v1** | **v2c** | **v3** [**auth** {**md5** | **sha**} {**clear** | **encrypted**} *auth-password* [**priv des56** {**clear** | **encrypted**} *priv-password*]]} [*access-list-name*] [**sdrowner**] [**systemowner**]

Example:

```
RP/0/RP0:hostname(config)# snmp-server user
noauthuser group_name v3
```

Configures a new user to an SNMP group.

Step 5 **commit**

Configuring SNMP Trap Notifications

This task explains how to configure the router to send SNMP trap notifications.

Procedure

Step 1 **configure**

Step 2 **snmp-server group** *name*{ **v1** | **v2** | **v3** } { **ipv4** | **ipv6** | **context** } [**read** *view*] [**write** *view*] [**notify** *view*] [*access-list-name*]

Example:

```
RP/0/RP0:hostname(config)# snmp-server group g1 v3 ipv4
view_name 1.3.6.1.2.1.1.5 included
```

Configures a new SNMP group or a table that maps SNMP users to SNMP views.

Step 3 **snmp-server user** *username groupname* {**v1** | **v2c** | **v3** [**auth** {**md5** | **sha**} {**clear** | **encrypted**} *auth-password* [**priv des56** {**clear** | **encrypted**} *priv-password*]]} [*access-list-name*] [**sdrowner**] [**systemowner**]

Example:

```
RP/0/RP0:hostname(config)# snmp-server user
noauthuser group_name v3
```

Configures a new user to an SNMP group.

Step 4 **snmp-server host** *address* [**traps**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*]

Example:

```
RP/0/RP0:hostname(config)# snmp-server host 12.26.25.61 traps version 3
noauth userV3noauth
```

Specifies SNMP trap notifications, the version of SNMP to use, the security level of the notifications, and the recipient (host) of the notifications.

Step 5 **snmp-server traps** [*notification-type*]

Example:

```
RP/0/RP0:hostname(config)# snmp-server traps bgp
```

Enables the sending of trap notifications and specifies the type of trap notifications to be sent.

- If a trap is not specified with the *notification-type* argument, all supported trap notifications are enabled on the router. To display which trap notifications are available on your router, enter the **snmp-server traps ?** command.

Step 6 **commit**

Step 7 (Optional) **show snmp host**

Example:

```
RP/0/RP0:hostname# show snmp host
```

Displays information about the configured SNMP notification recipient (host), port number, and security model.

Configure SNMP on a Node

This procedure enables the user to configure SNMP on a node ; the node now performs as an SNMP agent.

Procedure

Step 1 **configure**

Step 2 **snmp-server community public** *community-string* [**RO** | **RW**] [**SDROwner** | **SystemOwner**]

Example:

```
RP/0/RP0:hostname(config) # snmp-server community c1 RW SystemOwner
```

Configures the community access string to permit access to the Simple Network Management Protocol (SNMP). The **RW** keyword specifies read-write access and the authorized management stations can both, retrieve and modify MIB objects.

Step 3 **snmp-server traps otn**

Example:

```
RP/0/RP0:hostname(config) # snmp-server traps otn
```

Enables SNMP OTN traps.

Step 4 **snmp-server host** *host-address* **traps version**[**1** | **2c** | **3**] **public udp-port** *udp-port number*

Example:

```
RP/0/RP0:hostname(config) # snmp-server host 10.1.1.1 traps version 2c public udp-port 100
```

Configures the host address, SNMP version and the udp port number to which the notifications need to be sent.

Step 5 **commit**

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

This task explains how to set the system contact string, system location string, and system serial number of the SNMP agent.

Note The sequence in which you issue the **snmp-server** commands for this task does not matter.

Procedure

Step 1 **configure**

Step 2 (Optional) **snmp-server contact** *system-contact-string*

Example:

```
RP/0/RP0:hostname(config)# snmp-server contact
Dial System Operator at beeper # 27345
```

Sets the system contact string.

Step 3 (Optional) **snmp-server location** *system-location*

Example:

```
RP/0/RP0:hostname(config)# snmp-server location
Building 3/Room 214
```

Sets the system location string.

Step 4 (Optional) **snmp-server chassis-id** *serial-number*

Example:

```
RP/0/RP0:hostname(config)# snmp-server chassis-id 1234456
```

Sets the system serial number.

Step 5 **commit**

Defining the Maximum SNMP Agent Packet Size

This task shows how to configure the largest SNMP packet size permitted when the SNMP server is receiving a request or generating a reply.

Note The sequence in which you issue the **snmp-server** commands for this task does not matter.

Procedure

Step 1 **configure**

Step 2 (Optional) **snmp-server packetsize** *byte-count*

Example:

```
RP/0/RP0:hostname(config)# snmp-server packetsize 1024
```

Sets the maximum packet size.

Step 3 **commit**

Changing Notification Operation Values

After SNMP notifications have been enabled, you can specify a value other than the default for the source interface, message queue length, or retransmission interval.

This task explains how to specify a source interface for trap notifications, the message queue length for each host, and the retransmission interval.

Note The sequence in which you issue the **snmp-server** commands for this task does not matter.

Procedure

Step 1 **configure**

Step 2 (Optional) **snmp-server trap-source** *type interface-path-id*

Example:

```
RP/0/RP0:hostname(config)# snmp-server trap-source POS 0/0/1/0
```

Specifies a source interface for trap notifications.

Step 3 (Optional) **snmp-server queue-length** *length*

Example:

```
RP/0/RP0:hostname(config)# snmp-server queue-length 20
```

Establishes the message queue length for each notification.

Step 4 (Optional) **snmp-server trap-timeout** *seconds*

Example:

```
RP/0/RP0:hostname(config)# snmp-server trap-timeout 20
```

Defines how often to resend notifications on the retransmission queue.

Step 5 **commit**

Setting IP Precedence and DSCP Values

This task describes how to configure IP Precedence or IP DSCP for SNMP traffic.

Before you begin

SNMP must be configured.

Procedure

Step 1 **configure**

Step 2 Use one of the following commands:

- **snmp-server ipv4 precedence** [*value* | **critical** | **flash** | **flash-override** | **immediate** | **internet** | **network** | **priority** | **routine**]
- **snmp-server ipv4 dscp** [*value* | **af11...13** | **af21...23** | **af31...33** | **af41...43** | **cs1...cs7** | **default** | **ef**]

Example:

```
RP/0/RP0:hostname(config)# snmp-server dscp 24
```

Configures an IP precedence or IP DSCP value for SNMP traffic.

Step 3 **commit**

Displaying SNMP Context Mapping

The SNMP agent serves queries based on SNMP contexts created by the client features. There is a context mapping table. Each entry in the context mapping table includes a context name, the name of the feature that created the context, and the name of the specific instance of the feature.

Procedure

show snmp context-mapping

Example:

```
RP/0/RP0:hostname# show snmp context-mapping
```

Displays the SNMP context mapping table.

Monitoring Packet Loss

It is possible to monitor packet loss by configuring the generation of SNMP traps when packet loss exceeds a specified threshold. The configuration described in this task enables the creation of entries in the MIB tables of the EVENT-MIB. This can then be monitored for packet loss using SNMP GET operations.

Before you begin

Note Entries created in the EVENT-MIB MIB tables using the configuration described in this task cannot be altered using an SNMP SET.

Entries to the EVENT-MIB MIB tables created using an SNMP SET cannot be altered using the configuration described in this task.

Procedure

snmp-server mibs eventmib packet-loss *type interface-path-id* **falling** *lower-threshold* **interval** *sampling-interval* **rising** *upper-threshold*

Example:

```
RP/0/RP0:hostname(config)# snmp-server mibs eventmib packet-loss TenGigE 0/2/0/3 falling 1
 interval 5 rising 2
```

Generates SNMP EVENT-MIB traps for the interface when the packet loss exceeds the specified thresholds. Up to 100 interfaces can be monitored.

falling *lower-threshold* —Specifies the lower threshold. When packet loss between two intervals falls below this threshold and an mteTriggerRising trap was generated previously, a SNMP mteTriggerFalling trap is generated. This trap is not generated until the packet loss exceeds the upper threshold and then falls back below the lower threshold.

interval *sampling-interval* —Specifies how often packet loss statistics are polled. This is a value between 5 and 1440 minutes, in multiples of 5.

rising *upper-threshold* —Specifies the upper threshold. When packet loss between two intervals increases above this threshold, an SNMP mteTriggreRising trap is generated. This trap is not generated until the packet loss drops below the lower threshold and then rises above the upper threshold.

Configuring MIB Data to be Persistent

Many SNMP MIB definitions define arbitrary 32-bit indices for their object tables. MIB implementations often do a mapping from the MIB indices to some internal data structure that is keyed by some other set of data. In these MIB tables the data contained in the table are often other identifiers of the element being modelled. For example, in the ENTITY-MIB, entries in the entPhysicalTable are indexed by the 31-bit value, entPhysicalIndex, but the entities could also be identified by the entPhysicalName or a combination of the other objects in the table.

Because of the size of some MIB tables, significant processing is required to discover all the mappings from the 32-bit MIB indices to the other data which the network management station identifies the entry. For this reason, it may be necessary for some MIB indices to be persistent across process restarts, switchovers, or device reloads. The ENTITY-MIB entPhysicalTable and CISCO-CLASS-BASED-QOS-MIB are two such MIBs that often require index values to be persistent.

Also, because of query response times and CPU utilization during CISCO-CLASS-BASED-QOS-MIB statistics queries, it is desirable to cache service policy statistics.

Procedure

Step 1 (Optional) **snmp-server mibs cbqosmib persist**

Example:

```
RP/0/RP0:hostname(config)# snmp-server mibs cbqosmib persist
```

Enables persistent storage of the CISCO-CLASS-BASED-QOS-MIB data.

Step 2 (Optional) **snmp-server mibs cbqosmib cache refresh time** *time*

Example:

```
RP/0/RP0:hostname(config)# snmp-server mibs cbqosmib cache
refresh time 45
```

Enables QoS MIB caching with a specified cache refresh time.

Step 3 (Optional) **snmp-server mibs cbqosmib cache service-policy count** *count*

Example:

```
RP/0/RP0:hostname(config)# snmp-server mibs cbqosmib cache
service-policy count 50
```

Enables QoS MIB caching with a limited number of service policies to cache.

Step 4 (Optional) **snmp-server ifindex persist**

Example:

```
RP/0/RP0:hostname(config)# snmp-server ifindex persist
```

Enables ifIndex persistence on all interfaces that have entries in the ifIndex table of the IF-MIB. When enabled, this command retains the mapping between the ifName object values and the ifIndex object values persistent during reloads, allowing for consistent identification of specific interfaces using SNMP.

Configuring LinkUp and LinkDown Traps for a Subset of Interfaces

By specifying a regular expression to represent the interfaces for which you are interested in setting traps, you can enable or disable linkUp and linkDown traps for a large number of interfaces simultaneously.

Before you begin

SNMP must be configured.

Procedure

Step 1 **configure**

Step 2 **snmp-server interface subset** *subset-number* **regular-expression** *expression*

Example:

```
RP/0/RP0:hostname(config)# snmp-server interface subset 10
    regular-expression "^Gig[a-zA-Z]+[0-9/]+\."
RP/0/RP0:hostname(config-snmp-if-subset)#
```

Enters snmp-server interface mode for the interfaces identified by the regular expression.

The subset-number argument identifies the set of interfaces, and also assigns a priority to the subset in the event that an interface is included in more than one subset. Lower numbers have higher priority and their configuration takes precedent over interface subsets with higher numbers.

The *expression* argument must be entered surrounded by double quotes.

Refer to the *Understanding Regular Expressions, Special Characters, and Patterns* module in for more information regarding regular expressions.

Step 3 **notification linkupdown disable**

Example:

```
RP/0/RP0:hostname(config-snmp-if-subset)# notification linkupdown disable
```

Disables linkUp and linkDown traps for all interfaces being configured. To enable previously disabled interfaces, use the **no** form of this command.

Step 4 **commit**

Step 5 (Optional) **show snmp interface notification subset** *subset-number*

Example:

```
RP/0/RP0:hostname# show snmp interface notification subset 10
```

Displays the linkUp and linkDown notification status for all interfaces identified by the subset priority.

Step 6 (Optional) **show snmp interface notification regular-expression** *expression*

Example:

```
RP/0/RP0:hostname# show snmp interface notification
    regular-expression "^Gig[a-zA-Z]+[0-9/]+\."
```

Displays the linkUp and linkDown notification status for all interfaces identified by the regular expression.

Step 7 (Optional) **show snmp interface notification** *type interface-path-id*

Example:

```
RP/0/RP0:hostname# show snmp interface notification
    tengige 0/4/0/3.10
```

Displays the linkUp and linkDown notification status for the specified interface.

Generic IETF Traps

OTN supports the generic IETF traps listed in the following table.

call information	(Optional) Controls SNMP ISDN call information notifications, as defined in the CISCO-ISDN-MIB (enterprise 1.3.6.1.4.1.9.9.26.2). Notification types are: • demandNbrCallInformation (1) This notification is sent to the manager whenever a successful call clears, or a failed call attempt is determined to have ultimately failed. In the event that call retry is active, then this is after all retry attempts have failed. However, only one such notification is sent in between successful call attempts; subsequent call attempts do not generate notifications of this type. • demandNbrCallDetails (2) This notification is sent to the manager whenever a call connects, or clears, or a failed call attempt is determined to have ultimately failed. In the event that call retry is active, then this is after all retry attempts have failed. However, only one such notification is sent in between successful call attempts; subsequent call attempts do not generate notifications of this type.
chan-not-avail	(Optional) Controls SNMP ISDN channel-not-available notifications. ISDN PRI channel-not-available traps are generated when a requested DS-0 channel is not available, or when there is no modem available to take the incoming call. These notifications are available only for ISDN PRI interfaces.
ietf	(Optional) Controls the SNMP ISDN IETF traps.
isdnu-interface	(Optional) Controls SNMP ISDN U interface notifications.
layer2	(Optional) Controls SNMP ISDN Layer 2 transition notifications.

Usage Guidelines

SNMP notifications can be sent as traps or inform requests. This command enables both traps and inform requests for the specified notification types. ISDN notifications are defined in the CISCO-ISDN-MIB.my and CISCO-ISDNU-IF-MIB.my files, available on Cisco.com at http://www.cisco.com/public/mibs/v2/.

Availability of notifications will depend on your platform. To see what notifications are available, use the **snmp-server enable traps isdn ?** command.

If you do not enter an **snmp-server enable traps isdn** command, no notifications controlled by this command are sent. In order to configure the router to send these SNMP notifications, you must enter at least one

snmp-server enable traps isdn command. If you enter the command with no keywords, all notification types are enabled. If you enter the command with a keyword, only the notification type related to that keyword is enabled.

The **snmp-server enable traps snmp** command is used in conjunction with the **snmp-server host** command. Use the **snmp-server host** command to specify which host or hosts receive SNMP notifications. To send SNMP notifications, you must configure at least one snmp-server host command

Examples of IETF Traps

The following example shows how to determine what notification types are available on a Cisco AS5300 and then shows how to enable channel-not-available and Layer 2 informs:

```
NAS(config)# snmp-server enable traps isdn ?
call-information  Enable SNMP isdn call information traps
 chan-not-avail    Enable SNMP isdn channel not avail traps
ietf              Enable SNMP isdn ietf traps
 layer2            Enable SNMP isdn layer2 transition traps
<cr>
NAS(config)# snmp-server enable traps isdn chan-not-avail layer2
NAS(config)# snmp-server host myhost.cisco.com informs version 2c public isdn
```

SNMP Traps Supported in OTN

The following table lists the SNMP Traps Supported in OTN.

Table 4: SNMP Traps Supported in OTN

MIB Module
coiOtnIfOTUStatus
coiOtnIfODUStatus

MIB Supported in OTN

The following table lists the MIBs supported in OTN.

Table 5: MIBs Supported in OTN

MIB Module
RADIUS-AUTH-CLIENT-MIB
RADIUS-ACC-CLIENT-MIB
RADIUS-AUTH-CLIENT-MIB
CISCO-FLOW-MONITOR-MIB
CISCO-IP-CBR-METRICS-MIB
CISCO-FLOW-CLONE-MIB
ATM-MIB

MIB Module
ATM2-MIB
CISCO-ATM-EXT-MIB
IMA-MIB
CISCO-ATM-QOS-MIB
CISCO-OAM-MIB
IEEE8023-LAG-MIB
CISCO-CDP-MIB
ISIS-MIB
CISCO-CONFIG-MAN-MIB
CISCO-IPSEC-POLICY-MAP-MIB
CISCO-IPSEC-MIB
CISCO-IPSEC-FLOW-MONITOR-MIB
ETHERLIKE-MIB
CISCO-OTN-IF-MIB
CISCO-FLASH-MIB
FRAME-RELAY-DTE-MIB
CISCO-FRAME-RELAY-MIB
MFR-MIB
CISCO-CONFIG-COPY-MIB
CISCO-LICENSE-MGMT-MIB
CISCO-ENTITY-REDUNDANCY-MIB
CISCO-ENHANCED-MEMPOOL-MIB
CISCO-MEMORY-POOL-MIB
CISCO-PROCESS-MIB
CISCO-SYSLOG-MIB
CISCO-SYSTEM-MIB
CISCO-SELECTIVE-VRF-DOWNLOAD-MIB
CISCO-IETF-BFD-MIB
CISCO-NTP-MIB
IPv6-FORWARD-MIB
IP-FORWARD-MIB
RSVP-MIB

MIB Module
CISCO-TCP-MIB
TCP-MIB
UDP-MIB
BGP4-MIB
CISCO-BGP4-MIB
CISCO-HSRP-MIB
CISCO-HSRP-EXT-MIB
RFC2011-MIB
CISCO-MLD-SNOOPING-MIB
OSPF-TRAP-MIB
OSPF-MIB
CISCO-IETF-VRRP-07-MIB
VRRP-MIB
OSPFV3-MIB
CISCO-IP-STAT-MIB
CISCO-VLAN-IFTABLE-RELATIONSHIP-MIB
LLDP-MIB
IEEE8021-CFM-MIB
OAM-MIB
CISCO-IETF-VPLS-BGP-EXT-MIB
CISCO-IETF-PW-FR-MIB
CISCO-IETF-PW-MIB
CISCO-IETF-PW-MPLS-MIB
CISCO-IETF-PW-ENET-MIB
CISCO-IETF-VPLS-LDP-MIB
CISCO-IETF-VPLS-GENERIC-MIB
CISCO-TAP2-MIB
CISCO-USER-CONNECTION-TAP-MIB
CISCO-IP-TAP-MIB
CISCO-RTTMON-MIB
MPLS-LDP-STD-MIB
MPLS-LDP-GENERIC-STD-MIB

MIB Module
MPLS-LSR-STD-MIB
CISCO-MPLS-TE-STD-EXT-MIB
MPLS-TE-STD-MIB
CISCO-MPLS-TE-STD-EXT-MIB
CISCO-IETF-FRR-MIB
CISCO-IETF-MPLS-TE-P2MP-STD-MIB
MPLS-L3VPN-STD-MIB
DS1-MIB
CISCO-DS3-MIB
DS3-MIB
CISCO-FABRIC-C12K-MIB
CISCO-FABRIC-MCAST-APPL-MIB
CISCO-FABRIC-MCAST-MIB
CISCO-FABRIC-HFR-MIB
CISCO-CLASS-BASED-QOS-MIB
CISCO-CLASS-BASED-QOS-MIB
CISCO-TEST-MIB
CISCO-MIBD-ROUTE-TEST-MIB
CISCO-MIBD-INT-TEST-MIB
CISCO-ENTITY-ASSET-MIB
BRIDGE-MIB
CISCO-BULK-FILE-MIB
CISCO-BGP-POLICY-ACCOUNTING-MIB
CISCO-CONTEXT-MAPPING-MIB
CISCO-ENHANCED-IMAGE-MIB
ENTITY-MIB
ENTITY-STATE-MIB
CISCO-ENTITY-STATE-EXT-MIB
EVENT-MIB
DISMAN-EXPRESSION-MIB
CISCO-ENTITY-FRU-CONTROL-MIB
CISCO-FTP-CLIENT-MIB

MIB Module
IF-MIB
CISCO-IF-EXTENSION-MIB
IETF-TCP-MIB
RFC2465-MIB
IPV6-MIB
IETF-UDP-MIB
MAU-MIB
CISCO-MAU-EXT-MIB
CISCO-IETF-MSDP-MIB
CISCO-IETF-PIM-EXT-MIB
PIM-MIB
CISCO-PIM-MIB
CISCO-IETF-IPMROUTE-MIB
MGMDSTDMIB-MIB
IPV6-MLD-MIB
NOTIFICATION-LOG-MIB
CISCO-P2P-IF-MIB
CISCO-PING-MIB
CISCO-RF-MIB
CISCO-ENTITY-SENSOR-MIB
APS-MIB
CISCO-SONET-MIB
SONET-MIB
ATM-FORUM-MIB
ATM-FORUM-ADDR-REG
ATM-FORUM-SRVC-REG
CISCO-SC-MIB
HCNUM-TC
CISCO-CLASS-BASED-QOS-MIB
CISCO-SESS-BORDER-CTRLR-EVENT-MIB
CISCO-SESS-BORDER-CTRLR-CALL-STATS-MIB
mgmtrap

MIB Module
dbltrap
SNMPv2-MIB