MPLS Traffic Engineering MIB

First Published: May 22, 2001
Last Updated: November 1, 2006

The MPLS Traffic Engineering MIB enables Simple Network Management Protocol (SNMP) agent support in Cisco IOS software for Multiprotocol Label Switching (MPLS) traffic engineering (TE) management, as implemented in the MPLS Traffic Engineering MIB (MPLS TE MIB). The SNMP agent code operating in conjunction with the MPLS TE MIB enables a standardized, SNMP-based approach to be used in managing the MPLS TE features in Cisco IOS software.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for the MPLS Traffic Engineering MIB" section on page 31.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

CISCO SYSTEMS

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

Information About the MPLS Traffic Engineering MIB

This section describes the following:

MPLS Traffic Engineering MIB Cisco Implementation

The MPLS TE MIB is based on the Internet Engineering Task Force (IETF) draft MIB entitled *draft-ietf-mpls-te-mib-05.txt*, which includes objects describing features that support MPLS TE. This IETF draft MIB is revised occasionally and is becoming a standard. Accordingly, Cisco's implementation of the MPLS TE MIB is expected to track the evolution of the IETF draft MIB.

Slight differences between the IETF draft MIB and the implementation of the TE capabilities within Cisco IOS software require some minor translations between the MPLS TE MIB and the internal data structures of Cisco IOS software. These translations are made by the SNMP agent code that is installed and operating on various hosts within the network. This SNMP agent code, running in the background as a low priority process, provides a management interface to Cisco IOS software.

The SNMP objects defined in the MPLS TE MIB can be displayed by any standard SNMP utility. All MPLS TE MIB objects are based on the IETF draft MIB; thus, no specific Cisco SNMP application is required to support the functions and operations pertaining to the MPLS TE MIB.

MPLS Traffic Engineering Overview

MPLS TE capabilities in Cisco IOS software enable an MPLS backbone to replicate and expand upon the TE capabilities of Layer 2 ATM and Frame Relay networks.

TE capabilities are essential to effective management of service provider and Internet service provider (ISP) backbones. Such backbones must support high transmission capacities, and the networks incorporating backbones must be extremely resilient to link or node failures.

The MPLS TE facilities built into Cisco IOS software provide a feature-rich, integrated approach to managing the large volumes of traffic that typically flow through WANs. The MPLS TE facilities are integrated into Layer 3 network services, thereby optimizing the routing of IP traffic in the face of constraints imposed by existing backbone transmission capacities and network topologies.

Capabilities Supported by the MPLS Traffic Engineering MIB

The following functionality is supported in the MPLS Traffic Engineering MIB:

- The ability to generate and queue notification messages that signal changes in the operational status of MPLS TE tunnels.
- Extensions to existing SNMP commands that provide the ability to enable, disable, and configure notification messages for MPLS TE tunnels.

- The ability to specify the name or the IP address of a network management station (NMS) in the operating environment to which notification messages are to be sent.
- The ability to write notification configurations into nonvolatile memory.

Notification Generation Events

When MPLS TE notifications are enabled (see the **snmp-server enable traps (mpls)** command), notification messages relating to specific events within Cisco IOS software are generated and sent to a specified NMS in the network.

For example, an mplsTunnelUp notification is sent to an NMS when an MPLS TE tunnel is configured and the tunnel transitions from an operationally “down” state to an “up” state.

Conversely, an mplsTunnelDown notification is generated and sent to an NMS when an MPLS TE tunnel transitions from an operationally “up” state to a “down” state.

Finally, an mplstunnelRerouted notification is sent to the NMS under the following conditions:

- The signaling path of an existing MPLS TE tunnel fails for some reason and a new path option is signaled and placed into effect (that is, the tunnel is rerouted).
- The signaling path of an existing MPLS TE tunnel is fully operational, but a better path option can be signaled and placed into effect (that is, the tunnel can be reoptimized). This reoptimazation can be triggered by:
  - A timer
  - The issuance of an **mpls traffic-eng reoptimize** command
  - A configuration change that requires the resignaling of a tunnel

Path options are configurable parameters that you can use to specify the order of priority for establishing a new tunnel path. For example, you can create a tunnel head configuration and define any one of many path options numbered 1 through x, with “1” being the highest priority option and “x” being an unlimited number of lower priority path options. Thus, there is no limit to the number of path options that you can specify in this manner.

Notification Implementation

When an MPLS TE tunnel interface (or any other device interface, such as an Ethernet or Packet over SONET (POS) interface) transitions between an up and down state, an Interfaces MIB (ifMIB) link notification is generated. When such a notification occurs in an MPLS TE MIB environment, the interface is checked by software to determine if the notification is associated with an MPLS TE tunnel. If so, the interfaces MIB link notification is interlinked with the appropriate mplsTunnelUp or mplsTunnelDown notification to provide notification to the NMS regarding the operational event occurring on the tunnel interface. Hence, the generation of an Interfaces MIB link notification pertaining to an MPLS traffic engineering tunnel interface begets an appropriate mplsTunnelUp or mplsTunnelDown notification that is transmitted to the specified NMS.

An mplsTunnelRerouted notification is generated whenever the signaling path for an MPLS TE tunnel changes. However, software intelligence in the MPLS TE MIB prevents the reroute notification from being sent to the NMS when a TE tunnel transitions between an up or down state during an administrative or operational status check of the tunnel. Either an up/down notification or a reroute notification can be sent in this instance, but not both. This action prevents unnecessary traffic on the network.

Benefits of MPLS Traffic Engineering MIB

The MPLS Traffic Engineering MIB provides the following benefits:

- Provides a standards-based SNMP interface for retrieving information about MPLS TE.
- Provides information about the traffic flows on MPLS TE tunnels.
- Presents MPLS TE tunnel routes, including the configured route, the Interior Gateway Protocol (IGP) calculated route, and the actual route traversed.
- Provides information, in conjunction with the Interfaces MIB, about how a tunnel was rerouted in the event of a link failure.
- Provides information about the configured resources used for an MPLS TE tunnel.
- Supports the generation and queueing of notifications that call attention to major changes in the operational status of MPLS TE tunnels; forwards notification messages to a designated NMS for evaluation or action by network administrators.

MPLS Traffic Engineering MIB Layer Structure

The SNMP agent code supporting the MPLS TE MIB follows the existing model for such code in Cisco IOS software and is, in part, generated by the Cisco IOS tool set, based on the MIB source code.

The SNMP agent code, which has a layered structure similar to that of the MIB support code in Cisco IOS software, consists of four layers:

- Platform independent layer—This layer is generated primarily by the Cisco IOS MIB development tool set and incorporates platform and implementation independent functions. The Cisco IOS MIB development tool set creates a standard set of files associated with a MIB.
- Application interface layer—The functions, names, and template code for MIB objects in this layer are also generated by the Cisco IOS MIB development tool set.
- Application specific layer—This layer provides an interface between the application interface layer and the application program interface (API) and data structures layer and performs tasks needed to retrieve required information from Cisco IOS software, such as searching through data structures.
- API and data structures layer—This layer contains the data structures or APIs within Cisco IOS software that are retrieved or called in order to set or retrieve SNMP management information.

Restrictions for the MPLS Traffic Engineering MIB

The following restrictions apply to the MPLS TE MIB for Cisco IOS releases:

- Supports read-only (RO) permission for MIB objects.
- Contains no configuration support by means of SET functions, except for the mplsTunnelTrapEnable object (which has been made writable). Accordingly, the MPLS TE MIB contains indexing support for the Interfaces MIB.
- Supports only SNMP GET, GETNEXT, and GETBULK retrieval functions, except in the case of the mplsTunnelTrapEnable object (which has been made writable by means of SET functions).
- Contains no support for Guaranteed Bandwidth Traffic Engineering (GBTE) or Auto Bandwidth features.

Features and Technologies Related to MPLS Traffic Engineering MIB

The MPLS TE MIB feature is used in conjunction with the following:

- Standards-based SNMP network management application
- MPLS
- MPLS TE
- MPLS label switching router MIB (MPLS-LSR-MIB)

Supported Objects in the MPLS Traffic Engineering MIB

The MPLS TE MIB contains numerous tables and object definitions that provide read-only SNMP management support for the MPLS TE features in Cisco IOS software. The MPLS TE MIB conforms to Abstract Syntax Notation One (ASN.1), thus reflecting an idealized MPLS TE database.

Using any standard SNMP network management application, you can retrieve and display information from the MPLS TE MIB by using GET operations; similarly, you can traverse information in the MIB database for display by using GETNEXT operations.

The MPLS TE MIB tables and objects supported in Cisco IOS releases follow. Important MIB tables (those highlighted in bold type) are described briefly in accompanying text.

- mplsTunnelConfigured—Total number of tunnel configurations that are defined on this node.
- mplsTunnelActive—Total number of label-switched paths (LSPs) that are defined on this node.
- mplsTunnelTEDistProto—The IGP distribution protocol in use.
- mplsTunnelMaxHops—The maximum number of hops any given tunnel can utilize.
- mplsTunnelIndexNext—Unsupported; set to 0.
- **mplsTunnelTable**—Entries in this table with an instance of 0 and a source address of 0 represent tunnel head configurations. All other entries in this table represent instances of LSPs, both signaled and standby. If a tunnel instance is signaled, its operating status (operStatus) is set to “up” (1) and its instance corresponds to an active LSP.

  Tunnel configurations exist only on the tunnel head where the tunnel interface is defined. LSPs traverse the network and involve tunnel heads, tunnel midpoints, and tunnel tails.

  Pointers in the tunnel table refer to corresponding entries in other MIB tables. By using these pointers, you can find an entry in the mplsTunnelTable and follow a pointer to other tables for additional information. The pointers are the following: *mplsTunnelResourcePointer*, *mplsTunnelHopTableIndex*, *mplsTunnelARHopTableIndex*, and *mplsTunnelCHopTableIndex*.

  The tunnel table is indexed by tunnel ID, tunnel instance, tunnel source address, and tunnel destination address. The description of each entry has an alphabetic suffix (a), (b), or (c), if appropriate, to indicate the applicability of the entry

  a. For tunnel head configurations only
  b. For LSPs only
  c. For both tunnel head configurations and LSPs

  Following is a list and description of each entry.

  – mplsTunnelIndex—Same as tunnel ID (c).
  – mplsTunnelInstance—Tunnel instance of the LSP; 0 for head configurations (b).

- mplsTunnelIngressLSRId—Source IP address of the LSP; 0 for head configurations (b).
- mplsTunnelEgressLSRId—Destination IP address of the tunnel (c).
- mplsTunnelName—Command name for the tunnel interfaces (a).
- mplsTunnelDescr—Descriptive name for tunnel configurations and LSPs (c).
- mplsTunnelIsIf—Indicator of whether the entry represents an interface (c).
- mplsTunnelIfIndex—Index of the tunnel interface within the ifMIB (a).
- mplsTunnelXCPointer—(For midpoints only – no tails) Pointer for the LSP within the mplsXCTable of the MPLS LSR MIB (b).
- mplsTunnelSignallingProto—Signaling protocol used by tunnels (c).
- mplsTunnelSetupPrio—Setup priority of the tunnel (c).
- mplsTunnelHoldingPrio—Holding priority of the tunnel (c).
- mplsTunnelSessionAttributes—Session attributes (c).
- mplsTunnelOwner—Tunnel owner (c).
- mplsTunnelLocalProtectInUse—Not implemented (c).
- mplsTunnelResourcePointer—Pointer into the Resource Table (b).
- mplsTunnelInstancePriority—Not implemented (b).
- mplsTunnelHopTableIndex—Index into the Hop Table (a).
- mplsTunnelARHopTableIndex—Index into the AR Hop Table (b).
- mplsTunnelCHopTableIndex—Index into the C Hop Table (b).
- mplsTunnelPrimaryTimeUp—Amount of time, in seconds, that the current path has been up (a).
- mplsTunnelPathChanges—Number of times a tunnel has been resignalled (a).
- mplsTunnelLastPathChange—Amount of time, in seconds, since the last path resignaling occurred (a).
- mplsTunnelCreationTime—Time stamp when the tunnel was created (a).
- mplsTunnelStateTransitions—Number of times the tunnel has changed state (a).
- mplsTunnelIncludeAnyAffinity—Not implemented (a).
- mplsTunnelIncludeAllAffinity—Attribute bits that must be set for the tunnel to traverse a link (a).
- mplsTunnelExcludeAllAffinity—Attribute bits that must *not* be set for the tunnel to traverse a link (a).
- mplsTunnelPathInUse—Path option number being used for the tunnel's path. If no path option is active, this object will be 0 (a).
- mplsTunnelRole—Role of the tunnel on the router; that is, head, midpoint, or tail (c).
- mplsTunneltotalUptime—Amount of time, in seconds, that the tunnel has been operationally up (a).
- mplsTunnelInstanceUptime—Not implemented (b).
- mplsTunnelAdminStatus—Administrative status of a tunnel (c).
- mplsTunnelOperStatus—Actual operating status of a tunnel (c).
- mplsTunnelRowStatus—This object is used in conjunction with configuring a new tunnel. This object will always be seen as "active" (a).

  - mplsTunnelStorageType—Storage type of a tunnel entry (c).

- mplsTunnelHopListIndexNext—Next valid index to use as an index in the mplsTunnelHopTable.
- **mplsTunnelHopTable**—Entries in this table exist only for tunnel configurations and correspond to the path options defined for the tunnel. Two types of path options exist: *explicit* and *dynamic*. This table shows all hops listed in the explicit path options, while showing only the destination hop for dynamic path options. The tunnel hop table is indexed by tunnel ID, path option, and hop number.

  Following is a list and description of each table entry.

  - mplsTunnelHopListIndex—Primary index into the table.
  - mplsTunnelHopIndex—Secondary index into the table.
  - mplsTunnelHopAddrType—Indicates if the address of this hop is the type IPv4 or IPv6.
  - mplsTunnelHopIpv4Addr—The IPv4 address of this hop.
  - mplsTunnelHopIpv4PrefixLen—The prefix length of the IPv4 address.
  - mplsTunnelHopIpv6Addr—The IPv6 address of this hop.
  - mplsTunnelHopIpv6PrefixLen—The prefix length of the IPv6 address.
  - mplsTunnelHopAsNumber—This object will contain 0 or the AS number of the hop, depending on the value of mplsTunnelHopAddrType.
  - mplsTunnelHopLspId—This object will contain 0 or the LSPID of the tunnel, depending on the value of mplsTunnelHopAddrType.
  - mplsTunnelHopType—Denotes whether this tunnel hop is routed in a strict or loose fashion.
  - mplsTunnelHopRowStatus—This object is used in conjunction with the configuring of a new row in the table.
  - mplsTunnelHopStorageType—The storage type of this MIB object.

- mplsTunnelResourceIndexNext
- **mplsTunnelResourceTable**—Entries in this table correspond to the "Tspec" information displayed when you execute the **show mpls traffic-eng tunnels** command. These entries exist only for LSPs.

  The tunnel resource table is indexed by address and hop number. Following the *mplsTunnelResourcePointer* pointer from the tunnel table is the best way to retrieve information from this table.

  Following is a list and description of each table entry.

  - mplsTunnelResourceIndex—The primary index into this table.
  - mplsTunnelResourceMaxRate—The maximum rate, in bits per second, supported by this tunnel.
  - mplsTunnelResourceMeanRate—The mean rate, in bits per second, supported by this tunnel.
  - mplsTunnelResourceMaxBurstSize—The maximum burst size, in bytes, allowed by this tunnel.
  - mplsTunnelResourceRowStatus—This object is used in conjunction with the configuration of a new row in the table.
  - mplsTunnelResourceStorageType—The storage type of this MIB object.

- **mplsTunnelARHopTable**—Entries in this table correspond to the actual route taken by the tunnel, and whose route was successfully signaled by the network. The hops present in this table correspond to those present in the record route object (RRO) in Resource Reservation Protocol (RSVP). You can also display the information in this table by executing the **show mpls traffic-eng tunnels** command.

The actual route hop table is indexed by address and hop number. Following the *mplsTunnelARHopTableIndex* pointer from the tunnel table is the best way to retrieve information from this table. The entries in the table are listed and described below.

Following is a list and description of each table entry.

- mplsTunnelARHopListIndex—The primary index into this table.
- mplsTunnelARHopIndex—The secondary index into this table.
- mplsTunnelARHopIpv4Addr—The IPv4 address of this hop.
- mplsTunnelARHopIpv4PrefixLen—The prefix length of the IPv4 address.
- mplsTunnelARHopIpv6Addr—The IPv6 address of this hop.
- mplsTunnelARHopIpv6PrefixLen—The prefix length of the IPv6 address.
- mplsTunnelARHopAsNumber—This object will contain 0 or the AS number of the hop, depending on the value of mplsTunnelARHopAddrType.
- mplsTunnelARHopAddrType—The type of address for this MIB entry, either IPv4 or IPv6.
- mplsTunnelARHopType—Denotes whether this tunnel hop is routed in a strict or loose manner.

- **mplsTunnelCHopTable**—Entries in this table correspond to the explicit route object (ERO) in RSVP, which is used to signal the LSP. The list of hops in this table will contain those hops that are computed by the constraint-based shortest path first (SPF) algorithm. In those cases where “loose” hops are specified for the tunnel, this table will contain the hops that are “filled-in” between the loose hops to complete the path. If you specify a complete explicit path, the computed hop table matches your specified path.

  The computed hop table is indexed by address and hop number. Following the *mplsTunnelCHopTableIndex* pointer from the tunnel table is the best way to retrieve information from this table. The entries in the table are listed and described below.

  - mplsTunnelCHopListIndex—The primary index into this table.
  - mplsTunnelCHopIndex—The secondary index into this table.
  - mplsTunnelCHopAddrType—Indicates if the address of this hop is the type IPv4 or IPv6.
  - mplsTunnelCHopIpv4Addr—The IPv4 address of this hop.
  - mplsTunnelCHopIpv4PrefixLen—The prefix length of the IPv4 address.
  - mplsTunnelCHopIpv6Addr—The IPv6 address of this hop.
  - mplsTunnelCHopIpv6PrefixLen—The prefix length of the IPv6 address.
  - mplsTunnelCHopAsNumber—This object will contain 0 or the AS number of the hop, depending on the value of mplsTunnelHopAddrType.
  - mplsTunnelCHopType—Denotes whether this tunnel hop is routed in a strict or loose way.

- **mplsTunnelPerfTable**—The tunnel performance table, which augments the **mplsTunnelTable**, provides packet and byte counters for each tunnel. This table contains the following packet and byte counters:

  - mplsTunnelPerfPackets—This packet counter works only for tunnel heads.
  - mplsTunnelPerfHCPackets—This packet counter works only for tunnel heads.
  - mplsTunnelPerfErrors—This packet counter works only for tunnel heads.
  - mplsTunnelPerfBytes—This byte counter works for tunnel heads and tunnel midpoints, but not for tunnel tails.

  - mplsTunnelPerfHCBytes—This byte counter works for tunnel heads and tunnel midpoints, but not for tunnel tails.

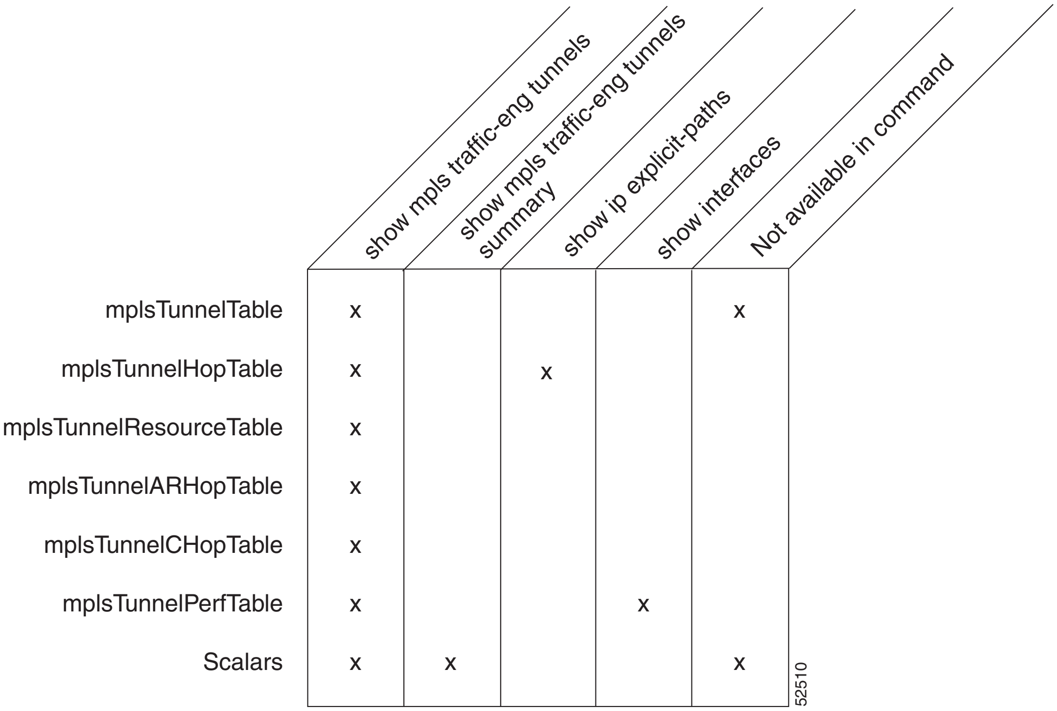
- mplsTunnelTrapEnable—The object type *mplsTunnelTrapEnable* is enhanced to be writable. Accordingly, if this object type is set to "TRUE," the following notifications are enabled, thus giving you the ability to monitor changes in the operational status of MPLS TE tunnels:
  - mplsTunnelUp
  - mplsTunnelDown
  - mplsTunnelRerouted

  If the *mplsTunnelTrapEnable* object is set to "FALSE," such operational status notifications are not generated. These notification functions are based on the definitions (mplsTeNotifications) contained in the IEFT draft document entitled *draft-ietf-mpls-te-mib-05.txt*.

CLI Access to MPLS Traffic Engineering MIB Information

Figure 1 shows commands that you can use to retrieve information from specific tables in the MPLS TE MIB. As noted in this figure, some information in the MPLS TE MIB is not retrievable by commands.

Figure 1 Commands for Retrieving MPLS TE MIB Information



	show mpls traffic-eng tunnels	show mpls traffic-eng tunnels summary	show ip explicit-paths	show interfaces	Not available in command
mplsTunnelTable	x				x
mplsTunnelHopTable	x		x		
mplsTunnelResourceTable	x				
mplsTunnelARHopTable	x				
mplsTunnelCHopTable	x				
mplsTunnelPerfTable	x			x	
Scalars	x	x			x

Retrieving Information from the MPLS Traffic Engineering MIB

This section describes how to efficiently retrieve information about TE tunnels. Such information can be useful in large networks that contain many TE tunnels.

Traverse across a single column of the *mplsTunnelTable*, such as *mplsTunnelName*. This action provides the indexes of every tunnel configuration, and any LSPs involving the host router. Using these indexes, you can perform a GET operation to retrieve information from any column and row of the *mplsTunnelTable*.

The *mplsTunnelTable* provides pointers to other tables for each tunnel. The column *mplsTunnelResourcePointer*, for example, provides an object ID (OID) that you can use to access resource allocation information in the *mplsTunnelResourceTabl*e. The columns *mplsTunnelHopTableIndex*, *mplsTunnelARHopTableIndex*, and *mplsTunnelCHopTableIndex* provide the primary index into the *mplsTunnelHopTable*, *mplsTunnelARHopTable*, and *mplsTunnelCHopTable*, respectively. By traversing the MPLS TE MIB in this manner using a hop table column and primary index, you can retrieve information pertaining to the hops of that tunnel configuration.

Because tunnels are treated as interfaces, the tunnel table column (*mplsTunnelIfInde*x) provides an index into the Interfaces MIB that you can use to retrieve interface-specific information about a tunnel.

How to Configure the MPLS Traffic Engineering MIB

This section contains the following tasks:

- Enabling the SNMP Agent to Help Manage Various MPLS TE Tunnel Characteristics of Tunnels on the Local Router, page 10 (required)
- Verifying the Status of the SNMP Agent, page 11 (optional)

Enabling the SNMP Agent to Help Manage Various MPLS TE Tunnel Characteristics of Tunnels on the Local Router

The SNMP agent for the MPLS TE MIB is disabled by default. To enable the SNMP agent for the MPLS TE MIB, perform the following steps.

SUMMARY STEPS

1. **telnet** *host*
2. **enable**
3. **show running-config**
4. **configure terminal**
5. **snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [**ipv6** *nacl*] [*access-list-number*]
6. **snmp-server enable traps** [*identification-type*] [*notification-option*]
7. **exit**
8. **write memory**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**telnet** *host* **Example:** `Router> telnet 192.172.172.172`	Telnets to the router identified by the specified IP address (represented as *xxx.xxx.xxx.xxx*).
Step 2	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 3	**show running-config** **Example:** `Router# show running-config`	Displays the running configuration to determine if an SNMP agent is already running. - If no SNMP information is displayed, go to Step 5. If any SNMP information is displayed, you can modify the information or change it as needed.
Step 4	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 5	**snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [**ipv6** *nacl*] [*access-list-number*] **Example:** `Router(config)# snmp-server community comaccess ro 4`	Enables the read-only (RO) community string.
Step 6	**snmp-server enable traps** [*identification-type*] [*notification-option*] **Example:** `Router(config)# snmp-server enable traps`	Enables an LSR to send SNMP notifications or informs to an SNMP host. **Note** This command is optional. After SNMP is enabled, all MIBs (not just the TE MIB) are available for the user to quer.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 8	**write memory** **Example:** `Router# write memory`	Writes the modified configuration to NVRAM, permanently saving the settings.

Verifying the Status of the SNMP Agent

To verify that the SNMP agent has been enabled on a host network device, perform the following steps.

Step 1 **telnet** *host*

Use this command to Telnet to the target device:

```
Router# telnet 192.172.172.172
```

Step 2 **enable**

Use this command to enable SNMP on the target device:

```
Router# enable
```

Step 3 **show running-config**

Use this command to display the running configuration on the target device and examine the output for displayed SNMP information.

```
Router# show running-config
.
.
.
snmp-server community public ro
snmp-server community private ro
```

Any **snmp-server** statement that appears in the output and takes the form shown here verifies that SNMP has been enabled on that device.

Configuration Examples for the MPLS Traffic Engineering MIB

This section contains the following configuration examples:

- Enabling the SNMP Agent to Help Manage Various MPLS TE Tunnel Characteristics of Tunnels on the Local Router: Example, page 12

Enabling the SNMP Agent to Help Manage Various MPLS TE Tunnel Characteristics of Tunnels on the Local Router: Example

The following example shows how to enable an SNMP agent on a host network device:

```
Router# configure terminal
Router(config)# snmp-server community snmp-community-string
```

The following example shows how to enable SNMPv1 and SNMPv2C. The configuration permits any SNMP agent to access all MPLS TE MIB objects with read-only permissions using the community string *publi*c.

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

The following example shows how to allow read-only access to all MPLS TE MIB objects relating to members of access list 4 that specify the *comaccess* community string. No other SNMP agents will have access to any MPLS TE MIB objects.

```
Router(config)# snmp-server community comaccess ro 4
```

Additional References

The following sections provide references related to the MPLS Traffic Engineering MIB.

Related Documents

Related Topic	Document Title
MPLS-based functionalities	- *MPLS Label Distribution Protocol (LDP)* - *MPLS Label Switching Router MIB* - *MPLS Scalability Enhancements for the LSC LSR* - *MPLS Scalability Enhancements for the ATM LSR* - *MPLS Traffic Engineering (TE)—Automatic Bandwidth Adjustment for MPLS TE Tunnels* - *MPLS Traffic Engineering (TE)—Scalability Enhancements* - *MPLS Class of Service Enhancements* - *RFC 2233 Interfaces MIB*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
MPLS TE MIB Interfaces MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2026	*The Internet Standards Process*

Technical Assistance

Description	Link
The Cisco Technical Support & Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Command Reference

This section documents modified commands only.

- **snmp-server community**
- **snmp-server enable traps (MPLS)**
- **snmp-server host**

snmp-server community

To set up the community access string to permit access to the Simple Network Management Protocol (SNMP), use the **snmp-server community** command in global configuration mode. To remove the specified community string, use the **no** form of this command.

snmp-server community *string* [**view** *view-name*] [**ro** | **rw**] [**ipv6** *nacl*] [*access-list-number*]

no snmp-server community *string*

Syntax Description

string	Community string that consists of 1 to 32 alphanumeric characters and functions much like a password, permitting access to SNMP. Blank spaces are not permitted in the community string. **Note** The @ sign is used for delimiting the context information.
view	(Optional) Specifies a previously defined view. The view defines the objects available to the SNMP community.
view-name	(Optional) Name of a previously defined view.
ro	(Optional) Specifies read-only access. Authorized management stations can retrieve only MIB objects.
rw	(Optional) Specifies read-write access. Authorized management stations can both retrieve and modify MIB objects.
ipv6	(Optional) Specifies an IPv6 named access list.
nacl	(Optional) IPv6 named access list.
access-list-number	(Optional) Integer from 1 to 99 that specifies a standard access list of IP addresses or a string (not to exceed 64 characters) that is the name of a standard access list of IP addresses allowed access to the SNMP agent. Alternatively, an integer from 1300 to 1999 that specifies a list of IP addresses in the expanded range of standard access list numbers that are allowed to use the community string to gain access to the SNMP agent.

Command Default

An SNMP community string permits read-only access to all objects.

Note If the **snmp-server community** command is not used during the SNMP configuration session, the command will automatically be added to the configuration after the **snmp host** command is used. In this case, the default password (*string*) for the **snmp-server community** command will be taken from the **snmp host** command.

Command Modes

Global configuration

Command History

Release	Modification
10.0	This command was introduced.
12.0(14)ST	This command was integrated into Cisco IOS Release 12.0(14)ST.

Release	Modification
12.0(17)S	This command was integrated into Cisco IOS Release 12.0(17)S.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.3(2)T	The access list values were enhanced to support the expanded range of standard access list values and to support named standard access lists.
12.0(27)S	The **ipv6** *nacl* keyword and argument pair was added to support assignment of IPv6 named access lists. This keyword and argument pair is not supported in Cisco IOS 12.2S releases.
12.3(14)T	The **ipv6** *nacl* keyword and argument pair was integrated into Cisco IOS Release 12.3(14)T to support assignment of IPv6 named access lists. This keyword and argument pair is not supported in Cisco IOS 12.2S releases.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.4(11)T	This command was integrated into Cisco IOS Release 12.4(11)T.
12.2(31)SB2	This command was integrated into Cisco IOS Release 12.2(31)SB2.

Usage Guidelines

The **no snmp-server** command disables all versions of SNMP (SNMPv1, SNMPv2C, SNMPv3).

The first **snmp-server** command that you enter enables all versions of SNMP.

To configure SNMP community strings for the MPLS LDP MIB, use the **snmp-server community** command on the host network management station (NMS).

The **snmp-server community** command can be used to specify only an IPv6 named access list, only an IPv4 access list, or both. For you to configure both IPv4 and IPv6 access lists, the IPv6 access list must appear first in the command statement.

Note The @ sign 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.

Examples

The following example shows how to set the read/write community string to newstring:

```
Router(config)# snmp-server community newstring rw
```

The following example shows how to allow read-only access for all objects to members of the standard named access list lmnop that specify the comaccess community string. No other SNMP managers have access to any objects.

```
Router(config)# snmp-server community comaccess ro lmnop
```

The following example shows how to assign the string comaccess to SNMP, allow read-only access, and specify that IP access list 4 can use the community string:

```
Router(config)# snmp-server community comaccess ro 4
```

The following example shows how to assign the string manager to SNMP and allow read-write access to the objects in the restricted view:

```
Router(config)# snmp-server community manager view restricted rw
```

The following example shows how to remove the community comaccess:

```
Router(config)# no snmp-server community comaccess
```

The following example shows how to disable all versions of SNMP:

```
Router(config)# no snmp-server
```

The following example shows how to configure an IPv6 access list named list1 and links an SNMP community string with this access list:

```
Router(config)# ipv6 access-list list1
Router(config-ipv6-acl)# permit ipv6 any any
Router(config-ipv6-acl)# exit
Router(config)# snmp-server community comaccess rw ipv6 list1
```

Related Commands

Command	Description
access-list	Configures the access list mechanism for filtering frames by protocol type or vendor code.
snmp-server enable traps	Enables the router to send SNMP notification messages to a designated network management workstation.
snmp-server host	Specifies the targeted recipient of an SNMP notification operation.
snmp-server view	Creates or updates a view entry.

snmp-server enable traps (MPLS)

To enable a label switch router (LSR) to send Simple Network Management Protocol (SNMP) notifications or informs to an SNMP host, use the **snmp-server enable traps** command in global configuration mode. To disable notifications or informs, use the **no** form of this command.

snmp-server enable traps [*notification-type*] [*notification-option*]

no snmp-server enable traps [*notification-type*] [*notification-option*]

Syntax Description

notification-type

(Optional) Specifies the particular type of SNMP notification(s) to be enabled on the LSR. If a notification type is not specified, all SNMP notifications applicable to the LSR are enabled and sent to the SNMP host. Any one or all of the following keywords can be specified in any combination as the *notification-type* (family name) in the **snmp-server enable traps** command:

- **bgp**—Sends Border Gateway Protocol (BGP) state change notifications.
- **config**—Sends configuration notifications.
- **entity**—Sends entity MIB modification notifications.
- **envmon**—Sends Cisco enterprise-specific environmental monitor notifications whenever certain environmental thresholds are exceeded. *Notification-option* arguments (below) can be specified in combination with this keyword.
- **frame-relay**—Sends Frame Relay notifications.
- **hsrp**—Sends Hot Standby Routing Protocol (HSRP) notifications.
- **isdn**—Sends ISDN notifications. *Notification-option* arguments (see below) can be specified in combination with this keyword.
- **repeater**—Sends Ethernet repeater (hub) notifications. *Notification-option* arguments (see below) can be specified in combination with this keyword.
- **rsvp**—Sends Resource Reservation Protocol (RSVP) notifications.
- **rtr**—Sends Service Assurance Agent/Response Time Reporter (RTR) notifications.
- **snmp** [**authentication**]—Sends RFC 1157 SNMP notifications. Using the **authentication** keyword produces the same effect as not using it. Both the **snmp-server enable traps snmp** and the **snmp-server enable traps snmp authentication** forms of this command globally enable the following SNMP notifications (or, if you are using the **no** form of the command, disables such notifications): **authenticationFailure**, **linkUp**, **linkDown**, and **warmstart**.
- **syslog**—Sends system error message (syslog) notifications. You can specify the level of messages to be sent using the **logging history level** command.

notification-type (continued)

- **mpls ldp**—Sends notifications about status changes in LDP sessions. Note that this keyword is specified as **mpls ldp**. This syntax, which the CLI interprets as a two-word construct, has been implemented in this manner to maintain consistency with other MPLS commands. *Notification-option* arguments (below) can be specified in combination with this keyword.
- **mpls traffic-eng**—Sends notifications about status changes in MPLS label distribution tunnels. This keyword is specified as **mpls traffic-eng**. This syntax, which the CLI interprets as a two-word construct, has been implemented in this manner to maintain consistency with other MPLS commands. *Notification-option* arguments (below) can be specified in combination with this keyword.

notification-option

(Optional) Defines the particular options associated with the specified *notification-type* that are to be enabled on the LSR.

- **envmon** [**voltage** | **shutdown** | **supply** | **fan** | **temperature**]

  When you specify the **envmon** keyword, you can enable any one or all of the following environmental notifications in any combination: **voltage**, **shutdown**, **supply**, **fan**, or **temperature**. If you do not specify an argument with the **envmon** keyword, all types of system environmental notifications are enabled on the LSR.
- **isdn** [**call-information** | **isdn u-interface**]

  When you specify the **isdn** keyword, you can use either the **call-information** argument (to enable an SNMP ISDN call information option for the ISDN MIB subsystem) or the **isdn u-interface** argument (to enable an SNMP ISDN U interface option for the ISDN U Interfaces MIB subsystem), or both. If you do not specify an argument with the **isdn** keyword, both types of isdn notifications are enabled on the LSR.
- **repeater** [**health** | **reset**]

  When you specify the **repeater** keyword, you can use either the **health** argument or the **reset** argument, or both (to enable the IETF Repeater Hub MIB [RFC 1516] notification). If you do not specify an argument with the **repeater** keyword, both types of notifications are enabled on the LSR.
- **mpls ldp** [**session-up** | **session-down** | **pv-limit** | **threshold**]

  When you specify the **mpls ldp** keyword, you can use any one or all of the following arguments in any combination to indicate status changes in LDP sessions: **session-up**, **session-down**, **pv-limit**, or **threshold**. If you do not specify an argument with the **mpls ldp** keyword, all four types of LDP session notifications are enabled on the LSR.
- **mpls traffic-eng** [**up** | **down** | **reroute**]

  When you specify the **mpls traffic-eng** keyword, you can use any one or all of the following arguments in any combination to enable the sending of notifications regarding status changes in MPLS label distribution tunnels: **up**, **down**, or **reroute**. If you do not specify an argument with the **mpls traffic-eng** keyword, all three types of tunnel notifications are enabled on the LSR.

Defaults

If you issue this command on an LSR without specifying any *notification-type* keywords, the default behavior of the LSR is to enable all notification types controlled by the command (some notification types cannot be controlled by means of this command).

Command Modes

Global configuration

Command History

Release	Modification
11.1	This command was introduced.
11.3	The **snmp-server enable traps snmp authentication** form of this command was introduced to replace the **snmp-server trap-authentication** command.
12.0(17)ST	The **mpls traffic-eng** keyword was added to define a class or family of specific SNMP notifications for use with the *notification-type* and *notification-option* parameters of the **snmp-server enable traps** command.
12.0(21)ST	The **mpls ldp** keyword was added to define a class or family of specific SNMP notifications for use with the *notification-type* and *notification-option* parameters of the **snmp-server enable traps** command.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.4(11)T	This command was integrated into Cisco IOS Release 12.4(11)T.
12.2(31)SB2	This command was integrated into Cisco IOS Release 12.2(31)SB2.

Usage Guidelines

To configure an LSR to send SNMP LDP notifications, you must issue at least one **snmp-server enable traps** command on the router.

To configure an LSR to send either notifications (traps) or informs to a designated network management station (NMS), you must issue the **snmp-server host** command on that device, using the keyword (**traps** or **informs**) that suits your purposes.

If you issue the **snmp-server enable traps** command without keywords, all SNMP notification types are enabled on the LSR. If you issue this command with specific keywords, only the notification types associated with those particular keywords are enabled on the LSR.

The **snmp-server enable traps** command is used in conjunction with the **snmp-server host** command. You use the latter command to specify the NMS host (or hosts) targeted as the recipient(s) of the SNMP notifications generated by SNMP-enabled LSRs in the network. To enable an LSR to send such notifications, you must issue at least one **snmp-server host** command on the LSR.

Examples

In the following example, the router is enabled to send all notifications to the host specified as myhost.cisco.com. The community string is defined as public.

```
Router(config)# snmp-server enable traps

Router(config)# snmp-server host myhost.cisco.com public
```

In the following example, the router is enabled to send Frame Relay and environmental monitor notifications to the host specified as myhost.cisco.com. The community string is defined as public:

```
Router(config)# snmp-server enable traps frame-relay

Router(config)# snmp-server enable traps envmon temperature

Router(config)# snmp-server host myhost.cisco.com public
```

In the following example, notifications are not sent to any host. BGP notifications are enabled for all hosts, but the only notifications enabled to be sent to a host are ISDN notifications (which are not enabled in this example).

```
Router(config)# snmp-server enable traps bgp

Router(config)# snmp-server host host1 public isdn
```

In the following example, the router is enabled to send all inform requests to the host specified as myhost.cisco.com. The community string is defined as public.

```
Router(config)# snmp-server enable traps

Router(config)# snmp-server host myhost.cisco.com informs version 2c public
```

In the following example, HSRP MIB notifications are sent to the host specified as myhost.cisco.com. The community string is defined as public.

```
Router(config)# snmp-server enable hsrp

Router(config)# snmp-server host myhost.cisco.com traps version 2c public hsrp
```

Related Commands

Command	Description
snmp-server host	Specifies the intended recipient of an SNMP notification (that is, the designated NMS workstation in the network).

snmp-server host

To specify the recipient of a Simple Network Management Protocol (SNMP) notification message, use the **snmp-server host** command in global configuration mode. To remove the specified host from the configuration, use the **no** form of this command.

snmp-server host {*hostname* | *ip-address*} [**vrf** *vrf-name*] [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*]

no snmp-server host {*hostname* | *ip-address*} [**vrf** *vrf-name*] [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*]

Syntax Description

hostname | *ip-address*	Name, IP address, or IPv6 address of the SNMP notification host. The SNMP notification host is typically a network management station (NMS or SNMP manager). This host is the recipient of the SNMP traps or informs.
vrf	(Optional) Specifies that a Virtual Private Network (VPN) routing and forwarding (VRF) instance should be used to send SNMP notifications.
vrf-name	(Optional) VPN VRF instance used to send SNMP notifications.
traps	(Optional) Specifies that notifications should be sent as traps. This is the default.
informs	(Optional) Specifies that notifications should be sent as informs.
version	(Optional) Version of the SNMP used to send the traps. The default is 1. If you use the **version** keyword, one of the following keywords must be specified: • **1**—SNMPv1. This option is not available with informs. • **2c**—SNMPv2C. • **3**—SNMPv3. The most secure model because it allows packet encryption with the **priv** keyword. The default is **noauth**. One of the following three optional security level keywords can follow the **3** keyword: – **auth**—Enables Message Digest 5 (MD5) and Secure Hash Algorithm (SHA) packet authentication. – **noauth**—Specifies that the noAuthNoPriv security level applies to this host. This is the default security level for SNMPv3. – **priv**—Enables Data Encryption Standard (DES) packet encryption (also called “privacy”).
community-string	Password-like community string sent with the notification operation. **Note** You can set this string using the **snmp-server host** command by itself, but Cisco recommends that you define the string using the **snmp-server community** command prior to using the **snmp-server host** command. **Note** The @ sign is used for delimiting the context information.

udp-port	(Optional) Specifies that SNMP notifications or informs are to be sent to an NMS host.
port	(Optional) User Datagram Protocol (UDP) port number of the NMS host. The default is 162.

notification-type	Optional) Type of notification to be sent to the host. If no type is specified, all available notifications are sent. The notification type can be one or more of the following keywords: • **bgp**—Sends Border Gateway Protocol (BGP) state change notifications. • **calltracker**—Sends Call Tracker call-start/call-end notifications. • **cef**—Sends Cisco Express Forwarding-related notifications. • **config**—Sends configuration change notifications. • **cpu**—Sends CPU-related notifications. • **director**—Sends DistributedDirector-related notifications. • **dspu**—Sends downstream physical unit (DSPU) notifications. • **eigrp**—Sends Enhanced Interior Gateway Routing Protocol (EIGRP) stuck-in-active (SIA) and neighbor authentication failure notifications. • **entity**—Sends Entity MIB modification notifications. • **envmon**—Sends Cisco enterprise-specific environmental monitor notifications when an environmental threshold is exceeded. • **flash**—Sends flash media insertion and removal notifications. • **frame-relay**—Sends Frame Relay notifications. • **hsrp**—Sends Hot Standby Routing Protocol (HSRP) notifications. • **iplocalpool**—Sends IP local pool notifications. • **ipmobile**—Sends Mobile IP notifications. • **ipsec**—Sends IP Security (IPsec) notifications. • **isdn**—Sends ISDN notifications. • **l2tun-pseudowire-status**—Sends pseudowire state change notifications. • **l2tun-session**—Sends Layer 2 tunneling session notifications. • **llc2**—Sends Logical Link Control, type 2 (LLC2) notifications. • **memory**—Sends memory pool and memory buffer pool notifications. • **mpls-ldp**—Sends Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP) notifications indicating status changes in LDP sessions. • **mpls-traffic-eng**—Sends MPLS traffic engineering notifications indicating changes in the status of MPLS traffic engineering tunnels. • **mpls-vpn**—Sends MPLS VPN notifications. • **ospf**—Sends Open Shortest Path First (OSPF) sham-link notifications. • **pim**—Sends Protocol Independent Multicast (PIM) notifications. • **repeater**—Sends standard repeater (hub) notifications. • **rsrb**—Sends remote source-route bridging (RSRB) notifications. • **rsvp**—Sends Resource Reservation Protocol (RSVP) notifications. • **rtr**—Sends Response Time Reporter (RTR) notifications. • **sdlc**—Sends Synchronous Data Link Control (SDLC) notifications.

- **sdllc**—Sends SDLC Logical Link Control (SDLLC) notifications.
- **snmp**—Sends any enabled RFC 1157 SNMP linkUp, linkDown, authenticationFailure, warmStart, and coldStart notifications.

Note To enable RFC 2233 compliant link up/down notifications, you should use the **snmp server link trap** command.

- **srp**—Sends Spatial Reuse Protocol (SRP) notifications.
- **stun**—Sends serial tunnel (STUN) notifications.
- **syslog**—Sends error message notifications (Cisco Syslog MIB). Specify the level of messages to be sent with the **logging history level** command.
- **tty**—Sends Cisco enterprise-specific notifications when a TCP connection closes.
- **voice**—Sends SNMP poor quality of voice traps, when used with the **snmp enable peer-trap poor qov** command.
- **vrrp**—Sends Virtual Router Redundancy Protocol (VRRP) notifications.
- **vsimaster**—Sends Virtual Switch Interface (VSI) Master notifications.
- **x25**—Sends X.25 event notifications.

Command Default

The router does not send any trap messages.

If you enter the **no snmp-server host** command with no keywords, traps (but not informs) are disabled to the host. To disable informs, use the **no snmp-server host informs** command.

Note If the community string is not defined using the **snmp-server community** command prior to using this command, the default form of the **snmp-server community** command will automatically be inserted into the configuration. The password (community-string) used for this automatic configuration of the **snmp-server community** will be the same as specified in the **snmp-server host** command. This automatic command insertion and use of passwords is the default behavior for Cisco IOS Release 12.0(3) and later releases.

Command Modes

Global configuration

Command History

Release	Modification
10.0	This command was introduced.
Cisco IOS Release 12 Mainline/T Train	
12.0(3)T	The **version 3** [**auth** | **noauth** | **priv**] syntax was added as part of the SNMPv3 Support feature. The **hsrp** notification-type keyword was added. The **voice** notification-type keyword was added.
12.1(3)T	The **calltracker** notification-type keyword was added for the Cisco AS5300 and AS5800 platforms.
12.2(2)T	The **vrf** *vrf-name* keyword and argument combination was added. The **ipmobile** notification-type keyword was added. Support for the **vsimaster** notification-type keyword was added for the Cisco 7200 and Cisco 7500 series.
12.2(4)T	The **pim** notification-type keyword was added. The **ipsec** notification-type keyword was added.
12.2(8)T	The **mpls-traffic-eng** notification-type keyword was added. The **director** notification-type keyword was added.
12.2(13)T	The **srp** notification-type keyword was added. The **mpls-ldp** notification-type keyword was added.
12.3(2)T	The **flash** notification-type keyword was added. The **l2tun-session** notification-type keyword was added.
12.3(4)T	The **cpu** notification-type keyword was added. The **memory** notification-type keyword was added. The **ospf** notification-type keyword was added.
12.3(8)T	The **iplocalpool** notification-type keyword was added for the Cisco 7200 and 7301 series routers.
12.3(11)T	The **vrrp** keyword was added.
12.3(14)T	Support for SNMP over IPv6 transport was integrated into Cisco IOS Release 12.3(14)T. Either an IP or IPv6 Internet address can be specified as the *hostname* argument. The **eigrp** notification-type keyword was added.
12.4(11)T	This command was integrated into Cisco IOS Release 12.4(11)T.
Cisco IOS Release 12.0S	
12.0(17)ST	The **mpls-traffic-eng** notification-type keyword was integrated into Cisco IOS Release 12.0(17)ST.
12.0(21)ST	The **mpls-ldp** notification-type keyword was integrated into Cisco IOS Release 12.0(21)ST.
12.0(22)S	All features in the Cisco IOS Release 12.0ST train were integrated into Cisco IOS Release 12.0(22)S. The **mpls-vpn** notification-type keyword was added.
12.0(23)S	The **l2tun-session** notification-type keyword was added.
12.0(26)S	The **memory** notification-type keyword was added.
12.0(27)S	Support for SNMP over IPv6 transport was added. Either an IP or IPv6 Internet address can be specified as the *hostname* argument. The **vrf** *vrf-name* keyword argument pair was integrated into Cisco IOS Release 12.0(27)S to support multiple Lightweight Directory Protocol contexts for VPNs.
12.0(31)S	The **l2tun-pseudowire-status** notification-type keyword was added.
Release 12.2S	

Release	Modification
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S.
12.2(25)S	The **cpu** notification-type keyword was added. The **memory** notification-type keyword was added.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(31)SB2	The **cef** notification-type keyword was added.

Usage Guidelines

SNMP notifications can be sent as traps or inform requests. Traps are unreliable because the receiver does not send acknowledgments when it receives traps. The sender cannot determine if the traps were received. However, an SNMP entity that receives an inform request acknowledges the message with a SNMP response protocol data unit (PDU). If the sender never receives the response, the inform request can be sent again. Thus, informs are more likely to reach their intended destination.

Compared to traps, informs consume more resources in the agent 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; an inform may be retried several times. The retries increase traffic and contribute to a higher overhead on the network.

If you do not enter a **snmp-server host** command, no notifications are sent. To configure the router to send SNMP notifications, you must enter at least one **snmp-server host** command. If you enter the command with no keywords, all trap types are enabled for the host.

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 informs** command for a host and then enter another **snmp-server host informs** 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, and the linkUpDown notifications are controlled by the **snmp trap link-status** command. These notification types do not require an **snmp-server enable** command.

A notification-type option's availability depends on the router type and Cisco IOS software features supported on the router. For example, the **envmon** notification type is available only if the environmental monitor is part of the system. To display what notification types are available on your system, use the command help **?** at the end of the **snmp-server host** command.

The **vrf** keyword allows you to specify the notifications being sent to a specified IP address over a specific VRF. The VRF defines a VPN membership of a customer so data is stored using the VPN.

Notification-Type Keywords

The *notification-type* keywords used in the **snmp-server host** command do not always match the keywords used in the corresponding **snmp-server enable traps** command. For example, the notification keyword applicable to MPLS traffic engineering tunnels is specified as **mpls-traffic-eng** (containing two hyphens and no intervening spaces). The corresponding parameter in the **snmp-server enable traps** command is specified as **mpls-traffic-eng** (containing an intervening space and a hyphen).

This syntax difference is necessary to ensure that the command-line interface (CLI) interprets the *notification-type* keyword of the **snmp-server host** command as a unified, single-word construct, which preserves the capability of the **snmp-server host** command to accept multiple *notification-type* keywords in the command line. The **snmp-server enable traps** commands, however, often use two-word constructs to provide hierarchical configuration options and to maintain consistency with the command syntax of related commands. Table 1 maps some examples of **snmp-server enable traps** commands to the keywords used in the **snmp-server host** command.

Table 1 Notification Keywords and Corresponding SNMP Enable Traps Commands

SNMP Enable Traps Command	SNMP Host Command Keyword
snmp-server enable traps l2tun session	**l2tun-session**
snmp-server enable traps mpls ldp	**mpls-ldp**
snmp-server enable traps mpls traffic-eng[1]	**mpls-traffic-eng**
snmp-server enable traps mpls vpn	**mpls-vpn**

[1] See the *Cisco IOS Multiprotocol Label Switching Command Reference* for documentation of this command.

Examples

If you want to configure a unique SNMP community string for traps but prevent SNMP polling access with this string, include an access list in the configuration. The following example shows how to name a community string comaccess and number an access list 10:

```
Router(config)# snmp-server community comaccess ro 10
Router(config)# snmp-server host 172.20.2.160 comaccess
Router(config)# access-list 10 deny any
```

Note The @ sign 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.

The following example shows how to send RFC 1157 SNMP traps to a host specified named myhost.cisco.com. Other traps are enabled, but only SNMP traps are sent because only **snmp** is specified in the **snmp-server host** command. The community string is defined as comaccess.

```
Router(config)# snmp-server enable traps
Router(config)# snmp-server host myhost.cisco.com comaccess snmp
```

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

```
Router(config)# snmp-server enable traps snmp
Router(config)# snmp-server enable traps envmon
Router(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 myhost.cisco.com using the community string public:

```
Router(config)# snmp-server enable traps
Router(config)# 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. The community string is defined as public.

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

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

```
Router(config)# snmp-server enable traps
Router(config)# snmp-server host myhost.cisco.com informs version 2c public
```

The following example shows how to send HSRP MIB informs to the host specified by the name myhost.cisco.com. The community string is defined as public.

```
Router(config)# snmp-server enable traps hsrp
Router(config)# snmp-server host myhost.cisco.com informs version 2c public hsrp
```

The following example shows how to send all SNMP notifications to example.com over the VRF named trap-vrf using the community string public:

```
Router(config)# snmp-server host example.com vrf trap-vrf public
```

The following example shows how to configure an IPv6 SNMP notification server with the IPv6 address 2001:0DB8:0000:ABCD:1 using the community string public:

```
Router(config)# snmp-server host 2001:0DB8:0000:ABCD:1 version 2c public udp-port 2012
```

The following example shows how to specify VRRP as the protocol using the community string public:

```
Router(config)# snmp-server enable traps vrrp
Router(config)# snmp-server host myhost.cisco.com traps version 2c public vrrp
```

The following example shows how to send all Cisco Express Forwarding informs to the notification receiver with the IP address 10.56.125.47 using the community string public.

```
Router(config)# snmp-server enable traps cef
Router(config)# snmp-server host 10.56.125.47 informs version 2c public cef
```

Command	Description
snmp-server enable peer-trap poor qov	Enables notifications of poor voice quality for applicable calls associated with a specific voice dial peer.
snmp-server enable traps	Enables SNMP notifications (traps and informs).
snmp-server informs	Specifies inform request options.
snmp-server link trap	Enables linkUp/linkDown SNMP traps, which are compliant with RFC 2233.
snmp-server trap-source	Specifies the interface (and hence the corresponding IP address) from which an SNMP trap should originate.
snmp-server trap-timeout	Defines how often to try resending trap messages on the retransmission queue.

Feature Information for the MPLS Traffic Engineering MIB

Table 2 lists the release history for this MIB.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 2 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 2 Feature Information for the MPLS Traffic Engineering MIB

Feature Name	Releases	Feature Information
MPLS Traffic Engineering MIB	12.0(17)S 12.0(17)ST 12.2(8)T 12.2(14)S 12.2(28)SB 12.2(31)SB2	The MPLS Traffic Engineering MIB feature enables the SNMP agent support in Cisco IOS software for MPLS TE management, as implemented in the MPLS TE MIB. In 12.0(17)S, this feature provided the ability to generate and queue SNMP notification messages that signal changes in the operational status of MPLS TE tunnels when you are using the MPLS TE MIB on Cisco 7500 series routers and Cisco 12000 series Internet routers. In 12.0(17)ST, support for SNMP traffic engineering notifications was extended to include Cisco 7500 series routers and Cisco 12000 series Internet routers. In 12.2(8)T, support for SNMP TE notifications was extended to include Cisco 7500 series routers. The **snmp-server host** command was modified. In 12.2(14)S, this feature was integrated. In 12.2(28)SB, this feature was integrated. In 12.2(31)SB2, this feature was integrated.

Glossary

affinity bits—An MPLS traffic engineering tunnel's requirements on the attributes of the links it will cross. The tunnel's affinity bits and affinity mask must match with the attributes of the various links carrying the tunnel.

call admission precedence—An MPLS traffic engineering tunnel with a higher priority will, if necessary, preempt an MPLS traffic engineering tunnel with a lower priority. An expected use is that tunnels that are harder to route will have a higher priority, and can preempt tunnels that are easier to route, on the assumption that those lower priority tunnels can find another path.

constraint-based routing—Procedures and protocols used to determine a route across a backbone taking into account resource requirements and resource availability, instead of simply using the shortest path.

flow—A traffic load entering the backbone at one point—point of presence (POP)—and leaving it from another that must be traffic engineered across the backbone. The traffic load will be carried across one or more LSP tunnels running from the entry POP to the exit POP.

headend—The LSR at which the tunnel originates. The tunnel's "head" or tunnel interface will reside at this LSR as well.

informs—A type of notification message that is more reliable than a conventional trap notification message because an informs message requires acknowledgment.

label—A short, fixed-length data construct that tells switching nodes how to forward data (packets or cells).

label-switched path (LSP) tunnel—A configured connection between two routers, using label switching to carry the packets.

LSP—label-switched path. A path that is followed by a labeled packet over several hops, starting at an ingress LSR and ending at an egress LSR.

LSR—label switch router. A Layer 3 router that forwards a packet based on the value of a label encapsulated in the packet.

MIB—Management Information Base. A database of network management information (consisting of MIB objects) that is used and maintained by a network management protocol such as SNMP. The value of a MIB object can be changed or retrieved using SNMP commands, usually by a GUI-based network management system. MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

MPLS—Multiprotocol Label Switching. Switching method that forwards IP traffic using a label. This label instructs the routers and the switches in the network where to forward the packets based on preestablished IP routing information.

notification (see traps)—A message sent by an SNMP agent to a network management station, console, or terminal to indicate that a significant event within Cisco IOS software has occurred.

NMS—network management station. An NMS is a powerful, well-equipped computer (typically an engineering workstation) that is used by a network administrator to communicate with other devices in the network. An NMS is typically used to manage network resources, gather statistics, and perform a variety of network administration and configuration tasks.

OSPF—Open Shortest Path First. A link-state routing protocol used for routing IP.

RSVP—Resource Reservation Protocol. Protocol for reserving network resources to provide quality of service (QoS) guarantees to application flows.

SNMP—Simple Network Management Protocol. A network management protocol used almost exclusively in TCP/IP networks. SNMP provides a means to monitor and control network devices, manage configurations, collect statistics, monitor performance, and ensure network security.

tailend—The downstream, receive end of a tunnel.

traffic engineering—Techniques and processes that cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods were used.

trap (see notification)—A message sent by an SNMP agent to a network management station, console, or terminal to indicate that a significant event within Cisco IOS software has occurred. Traps (notifications) are less reliable than inform requests, because the receiver of the trap does not send an acknowledgment of receipt; furthermore, the sender of the trap cannot determine if the trap was received.

VCI—virtual channel identifier. A 16-bit field in the header of an ATM cell. The VCI, together with the VPI, is used to identify the next network VCL as the cell passes through a series of ATM switches on its way to its final destination.

VCC—virtual channel connection. A VCC is a logical circuit consisting of VCLs that carries data between two endpoints in an ATM network. Sometimes called a virtual circuit connection.

VCL—virtual channel link. A VCL is the logical connection that exists between two adjacent switches in an ATM network.

VPI—virtual path identifier. An 8-bit field in the header of an ATM cell. The VPI, together with the VCI, is used to identify the next network VCL (see above) as the cell passes through a series of ATM switches on its way to its final destination.

Note See *Internetworking Terms and Acronyms* for terms not included in this glossary.

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.