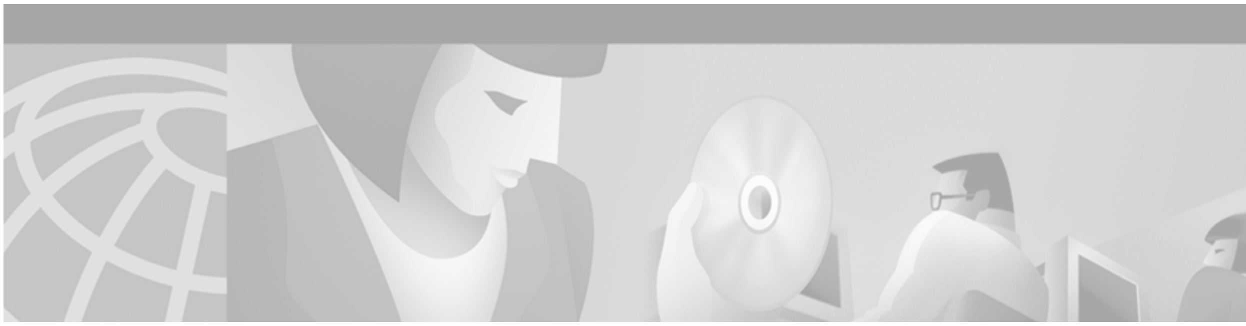

MPLS Label Distribution Protocol (LDP) MIB

Feature History

Release	Modification
12.0(11)ST	This feature was introduced to provide SNMP agent support when using the MPLS LDP MIB on Cisco 7200, Cisco 7500, and Cisco 12000 series routers.
12.2(2)T	This feature was added to this release to provide SNMP agent support when using the MPLS LDP MIB on Cisco 7200 and Cisco 7500 series routers.
12.0(21)ST	This feature was added to this release to provide SNMP agent and LDP notification support when using the MPLS LDP MIB on Cisco 7200, Cisco 7500, and Cisco 12000 series Internet routers.
12.2(11)S	This feature was integrated into Cisco IOS Release 12.2S.
12.0(22)S	This feature was integrated into Cisco IOS Release 12.0(22)S.

This document describes the Simple Network Management Protocol (SNMP) agent support provided in Cisco IOS for using the MPLS Label Distribution Protocol MIB (MPLS LDP MIB) on applicable Cisco IOS hardware platforms. The document contains the following sections:

Feature Overview

Multiprotocol label switching (MPLS) is a packet forwarding technology that uses a short, fixed-length value called a label in packets to determine the next hop for packet transport through an MPLS network by means of label switching routers (LSRs).

A fundamental MPLS principle is that LSRs in an MPLS network must agree on the definition of the labels being used for packet forwarding operations. Label agreement is achieved in an MPLS network by means of procedures defined in the Label Distribution Protocol (LDP).

LDP operations begin with a discovery (Hello) process during which an LDP entity (a local LSR) finds a cooperating LDP peer in the network and negotiates basic operating procedures between them. The recognition and identification of a peer by means of this discovery process results in an Hello adjacency, which represents the context within which label binding information is exchanged between the local LSR and its LDP peer. LDP functionality then creates an active LDP session between the two LSRs to effect the exchange of label binding information. The result of this process, when carried to completion with respect to all the LSRs in an MPLS network, is a label switched path (LSP), which constitutes an end-to-end packet transmission pathway between the communicating network devices.

By means of LDP, LSRs can collect, distribute, and release label binding information to other LSRs in an MPLS network, thereby enabling the hop-by-hop forwarding of packets in the network along normally routed paths.

The MPLS LDP MIB has been implemented to enable standard, SNMP-based network management of the label switching features in Cisco IOS. Providing this capability requires SNMP agent code to execute on a designated network management station (NMS) in the network. The NMS serves as the medium for user interaction with the network management objects in the MPLS LDP MIB.

The SNMP agent embodies a layered structure that is compatible with Cisco IOS and presents a network administrative and management interface to the objects in the MPLS LDP MIB and, thence, to the rich set of label switching capabilities supported by Cisco IOS.

By means of an SNMP agent, you can access MPLS LDP MIB objects using standard SNMP GET operations to accomplish a variety of network management tasks. All the objects in the MPLS LDP MIB follow the conventions defined in the Internet Engineering Task Force (IETF) draft MIB entitled *draft-ietf-mpls-ldp-mib-07.txt,* which defines network management objects in a structured and standardized manner. This draft MIB is continually being evolved toward the status of a standard. Accordingly, the MPLS LDP MIB will be implemented in a manner that tracks the evolution of this IETF document.

Slight differences that exist between the IETF draft MIB and the implementation of equivalent functions in Cisco IOS require some minor translations between the MPLS LDP MIB objects and the internal data structures of Cisco IOS. Such translations are accomplished by the SNMP agent, which runs in the background on the NMS workstation as a low priority process.

The extensive label switching capabilities supported in Cisco IOS provide an integrated approach to managing the large volumes of traffic carried by wide area networks (WANs). These capabilities are integrated into the Layer 3 network services, thus optimizing the routing of high volume traffic through Internet service provider backbones while, at the same time, ensuring the resiliency of the network to link or node failures.

This release of Cisco IOS supports the following functionality in relation to the MPLS LDP MIB:

- Generation and sending of event notification messages to signal changes in the status of LDP sessions.
- Enabling and disabling of event notification messages by means of extensions to existing SNMP CLI commands.

- Specification of the name or the IP address of an NMS workstation in the operating environment to which Cisco IOS event notification messages are to be sent to serve network administrative and management purposes.
- Storage of the configuration pertaining to an event notification message into non-volatile memory (NVRAM) of the NMS.

The structure of the MPLS LDP MIB conforms to Abstract Syntax Notation One (ASN.1), thereby forming a highly structured and idealized database of network management objects.

Using any standard SNMP application, you can retrieve and display information from the MPLS LDP MIB by means of standard SNMP GET operations; similarly, you can traverse and display information in the MIB by means of SNMP GETNEXT operations.

Note Due to the fact that the MPLS LDP MIB was not given an Internet Assigned Numbers Authority (IANA) Experimental OID at the time of its implementation, Cisco chose to implement the MIB under the Cisco Experimental OID number, as follows:
ciscoExperiment 1.3.6.1.4.1.9.10
mplsLdpMIB 1.3.6.1.4.1.9.10.65
In the event that the MPLS LDP MIB is assigned an IANA Experimental OID number, Cisco will deprecate all objects in the MIB under the ciscoExperimental OID and reposition the objects under the IANA Experimental OID.

Benefits of Using the MPLS LDP MIB

Representative benefits from using the MPLS LDP MIB in an MPLS environment include the following:

- Establishing LDP sessions between peer devices in an MPLS network
- Retrieving MIB parameters relating to the operation of LDP entities, such as:
  - Well known LDP discovery port
  - Maximum transmission unit (MTU)
  - Proposed KeepAlive timer interval
  - Loop detection
  - Session establishment thresholds
  - Range of VPI/VCI pairs to be used in forming labels
- Gathering statistics related to LDP operations, such as:
  - Count of the total established sessions for an LDP entity
  - Count of the total attempted sessions for an LDP entity
- Monitoring the time remaining for Hello adjacencies
- Monitoring the characteristics and status of LDP peers, such as:
  - Type of internetwork layer address of LDP peers
  - Actual internetwork layer address of LDP peers
  - Default MTU of the LDP peer
  - Number of seconds the LDP peer proposes as the value of the KeepAlive interval
  - Establishment of VPI/VCI label ranges to be made known to LDP peers

- Monitoring the characteristics and status of LDP sessions, such as:
  - Determining the LDP version being used by the LDP session
  - Determining the KeepAlive hold time remaining for an LDP session
  - Determining the state of an LDP session (whether the session is active or not)
  - Determining the range of VPI/VCI pairs to be used by an LDP session

Restrictions

This implementation of the MPLS LDP MIB is limited to read-only (RO) permission for MIB objects, except for MIB object *mplsLdpSessionUpDownTrapEnable*, which, for purposes of this release, has been extended to be writable by the SNMP agent.

Setting this object to a value of ‘true’ enables both the *mplsLdpSessionUp* and *mplsLdpSessionDown* notifications on the LSR; conversely, setting this object to a value of ‘false’ disables both of these notifications. The value of the *mplsLdpSessionUpDownTrapEnable* object is stored in NVRAM on the MPLS LDP MIB host.

For a description of notification events, see the section entitled Events Generating MPLS LDP MIB Notifications, page 8.

Most MPLS LDP MIB objects are set up automatically during the LDP peer discovery (Hello) process and the subsequent negotiation of parameters and establishment of LDP sessions between the LDP peers.

Related Features and Technologies

The MPLS LDP MIB is used in conjunction with the following related Cisco IOS technologies:

- Standards-based SNMP network management applications
- Multiprotocol label switching (MPLS)
- Label Distribution Protocol (LDP)

Related Documents

For descriptions of other MPLS-related functionality, refer to the following documentation:

- *MPLS Label Distribution Protocol*
- *MPLS Label Switching Router MIB*
- *MPLS Traffic Engineering MIB*
- *MPLS Scalability Enhancements for LSC and ATM LSR*
- *Automatic Bandwidth Adjustment for MPLS Label Distribution Tunnels*
- *Scalability Enhancements for MPLS Label Distribution*
- *MPLS Class of Service*
- *MPLS Label Distribution Access List Node Exclusion*
- *RFC 2233 Interfaces MIB*

Supported Platforms

The MPLS LDP MIB is supported on the following platforms in this Cisco IOS release:

- Cisco 7200 series routers
- Cisco 7500 series routers
- Cisco 12000 series routers

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

MPLS LDP MIB functionality is supported on the following Cisco IOS releases (listed in order of availability):

- Cisco IOS Release 12.0(11)ST—Provides SNMP agent support for the MPLS LDP MIB
- Cisco IOS Release 12.2(2)T—Provides SNMP agent support for the MPLS LDP MIB

- Cisco IOS Releases 12.0(21)ST, 12.2(11)S, 12.0(22)S—Provide SNMP agent support for the MPLS LDP MIB, as well as support for MPLS LDP MIB notifications

For descriptions of the MPLS LDP MIB and other supported MIBs and how to use them, see the Cisco MIB web site on Cisco Connection Online (CCO) at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

RFCs

The LDP implementation supporting the MPLS LDP MIB fully complies with the provisions of Section 10 of RFC 2026, which, in effect, states that the implementation of LDP is recommended for network devices that perform MPLS forwarding along normally routed paths, as determined by destination-based routing protocols.

Description of MPLS LDP MIB Elements

LDP operations related to an MPLS LDP MIB involve the following functional elements:

- LDP Entity—Relates to an instance of LDP for purposes of exchanging label spaces.
- LDP Peer—Refers to a remote LDP entity (that is, a non local LSR).
- LDP Session—Refers to an active LDP process between a local LSR and a remote LDP peer.
- Hello Adjacency—Refers to the result of an LDP discovery process which affirms the state of two LSRs in an MPLS network as being adjacent to each other (that is, as being LDP peers).

  An Hello Adjacency constitutes the working context between two LSRs in an MPLS network for purposes of exchanging label binding information.

These MPLS LDP MIB elements are briefly described under separate headings below.

In effect, the MPLS LDP MIB provides a network management database that supports real-time access to the various MIB objects within, reflecting the current state of MPLS LDP operations in the network. This network management information database is accessible by means of standard SNMP commands issued from an NMS in the MPLS/LDP operating environment.

The MPLS LDP MIB supports the following network management and administrative activities:

- Retrieving MPLS LDP MIB parameters pertaining to LDP operations.
- Monitoring the characteristics and the status of LDP peers.
- Monitoring the status of LDP sessions between LDP peers.
- Monitoring Hello adjacencies in the network.
- Gathering statistics regarding LDP sessions.

LDP Entities

An LDP entity is uniquely identified by an LDP identifier having the object name *mplsLdpEntityLdpId*. This object consists of the Router ID (four octets) and an interface number (two octets). The Router ID encodes an IP address assigned to the LSR. The interface number identifies a specific label space available within the LSR.

An LDP entity represents a label space that is targeted for distribution to an LDP peer. In the case of an interface-specific LDP entity, the label space is distributed to a single LDP peer by means of a single LDP session.

Conversely, a platform-wide LDP entity can be associated with multiple LDP peers. In this case, the label space is distributed to multiple LDP peers by means of a separate LDP session pertaining to each peer.

LDP Peers

If an LSR has a label space to advertise to another LSR, or to multiple LSRs, there would be one LDP session for each LSR receiving the label space information. The receiver of the label space information is referred to as an LDP peer.

Per-interface label spaces are advertised to a single LDP peer by means of a single LDP session. *Per-platform* label spaces are advertised to multiple LDP peers by means of multiple LDP sessions.

The possible existence of multiple per-platform LDP peers dictates not only that an LDP entity be identified by its unique LDP identifier, but also by its LDP Index. In this case, the label space is the same, but the LDP Index differentiates the LDP session over which the label space is distributed to multiple LDP peers.

LDP Sessions

LDP sessions exist between local entities and remote peers for purposes of distributing label spaces. There is always a one-to-one correspondence between an LDP peer and an LDP session. A single LDP session is a label distribution protocol instance that communicates across one or more network links with a single LDP peer. In the case of a platform-wide local LDP entity, there may be multiple LDP sessions and a corresponding number of remote LDP peers.

LDP Hello Adjacencies

An LDP session is an LDP instance that communicates across one or more network links to a peer protocol instance. An LDP Hello adjacency exists for each link on which LDP runs. Multiple link adjacencies exist whenever there are multiple links to the same LDP peer. In the case of a platform-wide label space, for example, there is a separate LDP peer/LDP session relationship for each LSR to which a label space may be advertised.

MPLS LDP MIB Object Categories

The MPLS LDP MIB contains numerous definitions of managed objects for the MPLS Label Distribution Protocol, as defined in the IETF draft document entitled *draft-ietf-mpls-ldp-07.txt*.

The managed objects in the MPLS LDP MIB are structured according to the following categories:

- MPLS LDP Textual Conventions
- MPLS LDP Objects
- MPLS Label Distribution Protocol Entity Objects
- LDP Entity Objects for Generic Labels
- LDP Entity Objects for ATM
- MPLS LDP Entity Configurable ATM Label Range Table

- MPLS Entity Objects for Frame Relay
- Frame Relay Label Range Components
- MPLS LDP Entity Statistics Table
- MPLS LDP Entity Peer Table
- MPLS LDP Hello Adjacency Table
- MPLS LDP Sessions Table
- MPLS LDP ATM Session Information
- MPLS LDP Frame Relay Session Information
- MPLS LDP Session Statistics Table
- Address Message/Address Withdraw Message Information
- MPLS LDP LIB Table
- MPLS LDP FEC Table
- Notifications
- Module Conformance Statement

Events Generating MPLS LDP MIB Notifications

When you enable MPLS LDP MIB notification functionality by issuing the **snmp-server enable traps mpls ldp** command, notification messages are generated and sent to a designated NMS in the network to signal the occurrence of specific events within Cisco IOS.

The MPLS LDP MIB objects involved in LDP status transitions and event notifications include the following:

- mplsLdpSessionUp—This message is generated when an LDP entity (a local LSR) establishes an LDP session with another LDP entity (an adjacent LDP peer in the network).
- mplsLdpSessionDown—This message is generated when an LDP session between a local LSR and its adjacent LDP peer is terminated.
- mplsLdpPathVectorLimitMismatch—This message is generated when a local LSR establishes an LDP session with its adjacent peer LSR, but the two LSRs have dissimilar path vector limits.

  The value of the path vector limit can range from 0 through 255; a value of “0” indicates that loop detection is off; any value other than zero up to 255 indicates that loop detection is on and, in addition, specifies the maximum number of hops through which an LDP message can pass before a loop condition in the network is sensed.

  It is recommended that all LDP-enabled routers in the network be configured with the same path vector limit. Accordingly, the *mplsLdpPathVectorLimitMismatch* object exists in the MPLS LDP MIB to provide a warning message to the NMS when two routers engaged in LDP operations have a dissimilar path vector limit.
- mplsLdpFailedInitSessionThresholdExceeded—This message is generated when a local LSR and an adjacent LDP peer attempt to set up an LDP session between them, but fail to do so after a specified number of attempts. The default number of attempts is 8. This default value is implemented in Cisco IOS and cannot be changed using either the CLI or an SNMP agent.

  Eight failed attempts to establish an LDP session between a local LSR and an LDP peer, due to any type of incompatibility between the devices, causes this notification message to be generated.

In general, Cisco routers support the same features across multiple platforms. Therefore, the most likely incompatibility to occur between Cisco LSRs is a mismatch of their respective ATM VPI/VCI label ranges.

For example, if you specify a range of valid labels for an LSR that does not overlap the range of its adjacent LDP peer, the routers will try eight times to create an LDP session between themselves before the *mplsLdpFailedInitSessionThresholdExceeded* notification is generated and sent to the NMS as an informational message.

Operationally, the LSRs whose label ranges do not overlap continue their attempt to create an LDP session between themselves after the eight retry threshold is exceeded. In such cases, the LDP threshold exceeded notification alerts the network administrator to the existence of a condition in the network that may warrant attention.

RFC 3036, *LDP Specification*, details the incompatibilities that can exist between Cisco routers and/or other vendor LSRs in an MPLS network. Among such incompatibilities, for example, are the following:

- Non-overlapping ATM VPI/VCI ranges (as noted above) or non-overlapping Frame-Relay DLCI ranges between LSRs attempting to set up an LDP session
- Unsupported label distribution method
- Dissimilar protocol data unit (PDU) size
- Dissimilar LDP feature support

VPN Contexts in the MPLS LDP MIB

Within an MPLS Border Gateway Protocol (BGP) 4 Virtual Private Network (VPN) environment, separate LDP processes can be created for each VPN. These processes and their associated data are called VPN contexts. Each context is independent from all others and contains data specific only to that context. The IETF MPLS-LDP MIB is capable only of showing information about a single context at one time.

Note This release supports a global VPN context only.

Configuration Tasks

This section describes the configuration tasks for using the MPLS LDP MIB:

- Enabling the SNMP Agent (required)
- Verifying the Status of the SNMP Agent (optional)

Enabling the SNMP Agent

By default, the SNMP agent for the MPLS LDP MIB is disabled. To enable the SNMP agent on the host NMS workstation, perform the steps in the following table:

	Command	Purpose
Step 1	`Prompt# telnet xxx.xxx.xxx.xxx`	Connects to the router at the specified IP address (represented as *xxx.xxx.xxx.xxx*).
Step 2	`Router# enable`	Establishes the enable mode.
Step 3	`Router# show running-config`	Displays the running configuration of the router to determine if an SNMP agent is already running on the device. If no SNMP information is displayed, continue with Step 4. If any SNMP information is displayed, you can modify the information or change it as desired.
Step 4	`Router# config terminal`	Establishes the global configuration mode.
Step 5	`Router(config)# snmp-server community xxxxxx RO`	Enables the read-only (*RO*) community string, where *xxxxxx* represents the read-only community string.
Step 6	`Router(config)# exit`	Exits the global configuration mode and returns to the privileged EXEC mode.
Step 7	`Router# write memory`	Writes the modified SNMP configuration into nonvolatile memory (NVRAM) of the router, permanently saving the SNMP settings.

Verifying the Status of the SNMP Agent

To verify that the SNMP agent has been enabled on the host NMS workstation, perform the steps in the following table:

Step 1 Connect to the target NMS workstation:

```
Router# telnet xxx.xxx.xxx.xxx
```

where `xxx.xxx.xxx.xxx` represents the IP address of the target NMS.

Step 2 Enable SNMP on the host NMS:

```
Router# enable
```

Step 3 Display the running configuration on the host NMS and examine the output for SNMP information:

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

The presence of any `snmp-server` statement in the output that takes the form shown above verifies that the SNMP agent has been enabled on the host NMS workstation.

Configuration Examples

The following example shows how to enable an SNMP agent on the host NMS:

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

The following example shows how to enable SNMPv1 and SNMPv2C on the host NMS. The configuration permits any SNMP agent to access all MPLS LDP MIB objects with read-only permission 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 LDP MIB objects relating to members of access list 4 that specify the *comaccess* community string. No other SNMP agents will have access to any of the MPLS LDP MIB objects.

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

Command Reference

This section documents CLI commands applicable to the MPLS LDP MIB for this Cisco IOS release.

- **snmp-server enable traps**
- **snmp-server host**

Other CLI commands used with the MPLS LDP MIB are documented in the Cisco IOS Release 12.2 command reference publications.

snmp-server enable traps

To enable an LSR to send SNMP notifications or informs to an SNMP host, use the **snmp-server enable traps** global configuration command. Use the **no** form of this command to disable notifications or informs.

snmp-server enable {**traps** | **informs**}[*notification-type*] [*notification-option*]

no snmp-server enable {**traps** | **informs**}[*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 (see 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 Integrated Services Digital Network (ISDN) notifications. *Notification-option* arguments (see below) can be specified in combination with this keyword. **repeater**—Sends Ethernet hub repeater 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. continued on the next page...

notification-type continued	**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 using the **no** form of the command, disables such notifications): **authenticationFailure**, **linkUp**, **linkDown**, or **warmstart**. **syslog—**Sends system error message (Syslog) notifications. You can specify the level of messages to be sent using the **logging history level** command. **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 2-word construct, has been implemented in this manner to maintain consistency with other MPLS commands. *Notification-option* arguments (see below) can be specified in combination with this keyword. **mpls traffic-eng**—Sends notifications about status changes in MPLS label distribution tunnels. Note that this keyword is specified as *mpls traffic-eng*. This syntax, which the CLI interprets as a 2-word construct, has been implemented in this manner to maintain consistency with other MPLS commands. *Notification-option* arguments (see 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. continued on next page...

notification-option	**repeater** [**health** | **reset**]
continued	When you specify the **repeater** keyword, you can use either the **health** argument (to enable the IETF Repeater Hub MIB (RFC 1516 notification) or the **reset** argument (to enable the IETF Repeater Hub MIB (RFC 1516 notification), or both. 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 tunnel notifications are enabled on the LSR.

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.
12.2(11)S	This command was integrated into Cisco IOS Release 12.2(11)S.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.

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 NMS, you must issue the **snmp-server host** command on that device using the desired 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*, using the community string defined as *public*:

```
snmp-server enable traps
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* using the community string *public*:

```
snmp-server enable traps frame-relay
snmp-server enable traps envmon temperature
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).

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

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

```
snmp-server enable traps
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* using the community string *public.*

```
snmp-server enable hsrp
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 an NMS workstation in the network as the intended recipient of SNMP notifications or informs, use the **snmp-server host** global configuration command. To disable the configuration of the NMS workstation as an SNMP host, use the **no** form of this command.

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

no snmp-server host *host-addr* [**traps** | **informs**]

Syntax Description

host-addr	Specifies the name or the IP address of the host NMS workstation on which the SNMP agent is running (thus serving as the recipient of SNMP notifications or informs).
traps	(Optional). Sends SNMP notifications to the specified NMS host. This is the default assumption of the **snmp-server host** command.
informs	(Optional). Sends SNMP informs to the specified NMS host.
version	(Optional). Indicates the SNMP version to be used in sending LDP notifications or informs to the NMS host. Version 3 is most secure, since it allows packet encryption by means of the **priv** keyword (see below). If you use the **version** keyword, one of the following arguments must also be specified: • **1**—SNMPv1. This option is not available when using the **informs** keyword. • **2c**—SNMPv2C. • **3**—SNMPv3. The following optional arguments can be used with the **version 3** keyword: – **auth**—Enables Message Digest 5 (MD5) and Secure Hash Algorithm (SHA) packet authentication. – **noauth**—Indicates the noAuthNoPriv security level. This argument is assumed as the default if the [**auth** | **noauth** | **priv**] keyword argument is not specified. – **priv**—Enables Data Encryption Standard (DES) packet encryption (also called “privacy” encryption).
community-string	The community string, functioning much like a password, is sent with the notification or informs operation. Although you can set this string using the **snmp-server host** command by itself, it is recommended that you define this string using the **snmp-server community** command prior to using the **snmp-server host** command.
udp-port *port*	UDP port number of the NMS host to which SNMP notifications or informs are to be sent. The default UDP port number is 162.

notification-type

(Optional). Specifies the particular type of SNMP notifications or informs to be sent to the NMS host. If no notification type is specified, all applicable SNMP notifications or informs are sent. Any one or more of the following can be specified as a keyword in this command:

- **bgp**—Sends Border Gateway Protocol (BGP) state change notifications.
- **config**—Sends configuration notifications.
- **dspu**—Sends downstream physical unit (DSPU) notifications.
- **entity**—Sends Entity MIB modification notifications.
- **envmon**—Sends Cisco enterprise-specific environmental monitor notifications when a specified system environmental threshold is exceeded.
- **frame-relay**—Sends Frame Relay notifications.
- **hsrp**—Sends Hot Standby Routing Protocol (HSRP) notifications.
- **isdn**—Sends Integrated Services Digital Network (ISDN) notifications.
- **llc2**—Sends Logical Link Control, Type 2 (LLC2) notifications.
- **mpls-ldp**—Sends notifications indicating status changes in LDP sessions. This keyword (embodying a dash) is seen as one word, enabling you to specify multiple keywords in the **snmp-server host** command (if you delimit each keyword with a space). The same parameter in the **snmp-server enable traps** command is specified as *mpls ldp* (embodying a space), which is seen as two words in the *notification-type* and *notification-option* parameters in the **snmp-server enable traps** command for consistency with other MPLS commands.
- **mpls-traffic-eng**—Sends notifications indicating status changes in label distribution tunnels. This keyword (embodying dashes) is seen as one word, enabling you to specify multiple keywords in the **snmp-server host** command (if you delimit each keyword with a space). The same parameter in the **snmp-server enable traps** command is specified as *mpls traffic-eng* (with an embedded space and dash), which is seen as two words in the *notification-type* and *notification-option* parameters in the **snmp-server enable traps** command for consistency with other MPLS commands.
- **repeater**—Sends standard repeater (hub) notifications.
- **rsrb**—Sends remote source-route bridging (RSRB) notifications.
- **rsvp**—Sends Resource Reservation Protocol (RSVP) notifications.
- **rtr**—Sends SA Agent (RTR) notifications.
- **sdlc**—Sends Synchronous Data Link Control (SDLC) notifications.
- **sdllc**—Sends SDLLC notifications.
- **snmp**—Sends SNMP notifications (as defined in RFC 1157).
- **stun**—Sends serial tunnel (STUN) notifications.
- **syslog**—Sends system error message (Syslog) notifications. You can specify the level of messages to be sent using the **logging history level** command.
- **tty**—Sends Cisco enterprise-specific notifications when a Transmission Control Protocol (TCP) connection terminates.
- **x25**—Sends X.25 event notifications.

Defaults

This command is disabled by default, in which case, no SNMP notifications are sent.

If you enter this command without keywords, the default is to send all notification types to the NMS host. No informs will be sent to the host.

If no **version** keyword is specified, the default is version 1. Issuing the **no snmp-server host** command without keywords disables notifications, but not informs, to the NMS 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 issuing the **snmp-server host** command, the default form of the **snmp-server community** command is automatically 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 is the default behavior for Cisco IOS Release 12.0(3) and later.

Command Modes

Global configuration

Command History

Release	Modification
10.0	This command was introduced.
12.0(17)ST	The **mpls-traffic-eng** keyword was added as a *notification-type* parameter in the **snmp-server host** command to enable sending traffic engineering notifications reflecting status changes in label distribution tunnels.
12.0(21)ST	The **mpls-ldp** keyword was added as a *notification-type* parameter in the **snmp-server host** command to enable sending LDP notifications reflecting status changes in LDP sessions.
12.2(11)S	This command was integrated into Cisco IOS Release 12.2(11)S.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.

Usage Guidelines

To configure an LSR to send SNMP notifications to an NMS, you must enter at least one **snmp-server host** command on the LSR.

If you issue the **snmp-server host** command without keywords, all SNMP notification types are enabled for the specified NMS host. If you issue this command with specific keywords, only the notification types associated with those particular keywords are enabled for the NMS host.

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

When multiple **snmp-server host** commands are issued for the same NMS host and notification type (trap or inform request), each succeeding such command issued overwrites the previous command. For example, if you issue an **snmp-server host inform** command for an NMS host, and then issue another **snmp-server host inform** command for the same NMS host, the second command overrides the first command.

The **snmp-server host** command is used in conjunction with the **snmp-server enable** command. You use the **snmp-server enable** command to specify which SNMP notifications are to be sent globally. For an NMS host to receive most SNMP notifications, at lease one **snmp-server enable** command and the **snmp-server host** command for that NMS host must be enabled on the LSR.

Examples

If you want to configure a unique SNMP community string for notifications, but you want to prevent SNMP polling access with this particular string, the configuration should include an access-list. In the following example, the community string is named *comaccess* and the access list is numbered 10:

```
snmp-server community comaccess ro 10
snmp-server host 172.20.2.160 comaccess
access-list 10 deny any
```

In the following example, SNMP notifications are sent to the NMS host specified as *myhost.cisco.com*. The community string is defined as *comaccess*.

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

In the following example, SNMP and the Cisco environmental monitor (envmon) enterprise-specific notifications are sent to the NMS host identified by IP address *172.30.2.160*:

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

In the following example, the LSR is enabled to send all notifications to the SNMP host identified as *myhost.cisco.com* using the community string *public*:

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

In the following example, notifications will not be sent to any SNMP host. The BGP notifications are enabled for all hosts, but only the ISDN notifications are enabled for sending to the host NMS.

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

In the following example, the LSR is enabled to send all inform requests to the NMS host specified as *myhost.cisco.com* using the community string *public*:

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

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

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

Related Commands

Command	Description
snmp-server enable traps	Enables the LSR on which this command is executed to send SNMP notifications to a designated NMS host.

Glossary

affinity bits—an MPLS label distribution tunnel's requirements on the attributes of the links it will cross. The tunnel's affinity bits and affinity mask must match up with the attributes of the various links carrying the tunnel.

call admission precedence—an MPLS label distribution tunnel with a higher priority will, if necessary, preempt an MPLS label distribution 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.

head-end—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, since the informs message notification requires acknowledgment, while a trap notification does not.

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. **label-switched path (LSP)**—A sequence of hops (R0...Rn) in which a packet travels from R0 to Rn through label switching mechanisms. A -switched path can be chosen dynamically, based on normal routing mechanisms, or through configuration.

Management Information Base—See MIB.

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 means of 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. An emerging industry standard that defines support for MPLS forwarding of packets along normally routed paths (sometimes called MPLS hop-by-hop forwarding).

Multiprotocol Label Switching label distribution—MPLS label distribution. A constraint-based routing algorithm for routing LSP tunnels.

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 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 (OSPF). A link state routing protocol used for routing IP.

RSVP—Resource Reservation Protocol. Protocol for reserving network resources to provide Quality of Service guarantees to application flows.

Simple Network Management Protocol—See SNMP.

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.

tail-end—The downstream, receive end of a tunnel.

label distribution—The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been 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 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 (see below), is used to identify the next network VCL (see below) 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 (see below) that carries data between two end points 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 (see above), 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.