



MPLS Label Distribution Protocol (LDP) MIB

Multiprotocol label switching (MPLS) is a packet forwarding methodology that uses a short, fixed-length value (called a label) in packets to enable the determination of the next hop for transporting packets through an MPLS network. An underlying MPLS concept is that two label switching routers (LSRs) must agree on the definition of the labels used to forward network traffic between and through them. This common understanding of labels is achieved through a set of procedures embodied in the Label Distribution Protocol (LDP). LDP enables an LSR to inform other LSRs of the label bindings it has made, thereby distributing label binding information to peer devices for the purpose of supporting hop-by-hop forwarding along normally routed paths.

To use LDP to best advantage in an MPLS network, the MPLS Label Distribution Protocol MIB (MPLS LDP MIB) has been implemented in conjunction with MPLS and LDP. Designed as a network management aid, the MPLS LDP MIB is based on an Internet Engineering Task Force (IETF) draft that defines objects in a structured and standardized label switching database.

The information in the MPLS LDP MIB is accessible by means of any network management utility that supports the Simple Network Management Protocol (SNMP). The SNMP-based code in a network management utility incorporates a layered structure for supporting the MPLS LDP MIB that is similar to that built into Cisco IOS software for supporting MIBs.

This document includes the following sections:

- [Feature Overview, page 2](#)
- [Benefits Derived from MPLS LDP MIBs, page 5](#)
- [Restrictions, page 6](#)
- [Supported Platforms, page 6](#)
- [Supported Standards, MIBs, and RFCs, page 6](#)
- [Configuration Tasks, page 6](#)
- [Configuration Examples, page 8](#)
- [Command Reference, page 8](#)
- [Glossary, page 11](#)

Feature Overview

The notation used in the MPLS LDP MIB adheres to the conventions defined in the Abstract System Notation One (ASN.1) standard, which defines an Open System Interconnection (OSI) language used in describing data types independently from particular computer structures and presentation techniques. Each object in the MPLS LDP MIB incorporates a DESCRIPTION field that describes the object's meaning and usage, which, together with other object characteristics, such as SYNTAX, MAX-ACCESS, and INDEX, provide sufficient information to meet the following needs:

- Developing SNMP-based network management applications
- Providing documentation
- Supporting testing

Thus, the MPLS LDP MIB is an idealized label switching data base that provides an effective management infrastructure for using LDP in an MPLS network. For example, the MPLS LDP MIB enables network administrators to:

- Monitor and control network devices
- Measure network performance
- Collect network statistics

The entries (objects) in an MPLS LDP MIB can be accessed in the following ways:

- By a network administrator—Using an SNMP-based network management utility running on a host network management station (NMS) in the network, a network administrator can retrieve information from the MPLS LDP MIB for display by means of standard SNMP GET operations.

The SNMP network management utility runs as a low priority process in the background, and the size and structure of the MPLS LDP MIB influences response time in retrieving information from the label switching data base.

- By LDP through the peer discovery process—The LDP peer discovery process enables LSRs to discover peers in an MPLS network and to establish LDP sessions with those peers for the purpose of exchanging label binding information. Thus, LSRs can collect, distribute, and release label prefix binding information to other LSRs in an MPLS network.

Description of MPLS LDP MIB Elements

LDP operations related to MPLS LDP MIBs involve the following functional elements:

- LDP Entity—Relates to an instance of LDP for the purpose of exchanging label spaces.
- LDP Peers—Refers to remote LDP entities.
- LDP Sessions—Refers to a conversation between LDP peers.
- Hello Adjacency—Refers to the result of a discovery process for determining network neighbors. An Hello Adjacency constitutes the working context between network peers for the purpose of exchanging label spaces .

LDP operations begin with a discovery (Hello) process during which an LDP entity finds cooperating LDP peers and negotiates basic operating parameters between the peers. The recognition of a peer by means of this discovery process results in an Hello adjacency, which is the context within which label spaces are exchanged between the cooperating peers. LDP functionality then establishes LDP sessions with Peers to exchange label binding information.

The MPLS LDP MIB provides a structure for managing the elements described above, enabling real-time access to the various states and counters comprising the LDP operating environment. Thus, the MPLS LDP MIB provides a vehicle that enables SNMP agent code running on network management stations to:

- Read MPLS LDP MIB parameters (LDP entities) related to LDP operation. The objects in the MPLS LDP MIB can be read by any standard SNMP network management utility.
- Monitor the characteristics and the status of LDP peers.
- Monitor the characteristics and status of LDP sessions.
- Monitor Hello adjacencies.
- Gather statistics from LDP sessions regarding LDP operation.

Functional Description of MPLS LDP MIB Elements

This section briefly describes the functions of the MPLS LDP MIB elements cited in the preceding section.

LDP Entities

An LDP entity is uniquely identified by an LDP identifier that takes 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 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. In contrast, 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 for 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, while per-platform label spaces are advertised to multiple LDP peers by means of multiple LDP sessions.

The possibility of the 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 the purpose of transferring label spaces. There is always a one-to-one relationship between an LDP peer and an LDP session. A single LDP session is a protocol instance that communicates across one or more links to a single peer protocol instance. In the case of a platform-wide local LDP entity, there may be multiple sessions and a corresponding number of remote peers.

LDP Hello Adjacencies

An LDP session is an LDP instance that communicates across one or more links to a peer protocol instance. An LDP Hello adjacency exists for each link on which LDP runs. Multiple link adjacencies exist when 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-05.txt*. This IETF draft document defines and arranges the objects in an MPLS LDP MIB 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

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.

Benefits Derived from MPLS LDP MIBs

The following is representative of the functions and benefits available to you by means of an MPLS LDP MIB:

- Reading MIB parameters related 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 the status of LDP peers, such as:
 - Type of the 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 ranges of labels to be made known to LDP peers
- Monitoring the characteristics and the status of LDP sessions, such as:
 - Version of LDP being used by the LDP session
 - KeepAlive hold time remaining for the LDP session
 - Indication of whether the LDP session is “active” or “passive”
 - Current state of the LDP session
 - Range of VPI/VCI pairs to be used by the LDP session
- Creating LDP sessions

Restrictions

This implementation of the MPLS LDP MIB for Cisco IOS Release 12.2(2)T is limited to read-only (RO) permission for MIB objects.

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

Supported Platforms

The MPLS LDP MIB is supported on the following platforms:

- Cisco 7200 series routers
- Cisco 7500 series routers

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

The MPLS LDP MIB is supported on Cisco IOS Release 12.0(11)ST and Cisco IOS Release 12.2(2)T.

For descriptions of supported MIBs and how to use them, see the Cisco MIB web site on Cisco Connection Online (CCO) at <http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>.

RFCs

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

Configuration Tasks

This section describes the following MPLS LDP MIB configuration tasks:

- [Enabling the SNMP Agent](#) (Required)
- [Verifying the Enabling of the SNMP Agent](#) (Optional)

Enabling the SNMP Agent

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

	Command	Purpose
Step 1	Prompt# telnet xxx.xxx.xxx.xxx	Telnets to the router identified by the specified IP address (represented as <i>xxx.xxx.xxx.xxx</i>).
Step 2	Router# enable	Enters the enable mode.
Step 3	Router# show running-config	Displays the running configuration to determine if an SNMP agent is already running. 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 needed.
Step 4	Router# config terminal	Enters the global configuration mode.
Step 5	Router(config)# snmp-server community xxxxxx RO	Enables the read-only (<i>RO</i>) community string, where <i>xxxxxx</i> represents the read-only community string
Step 6	Router(config)# snmp-server community xxxxxx RW	Enables the read-write (<i>RW</i>) community string, where <i>xxxxxx</i> represents the read-write community string.
Step 7	Router(config)# exit	Exits the global configuration mode and returns you to the privileged EXEC mode.
Step 8	Router# write memory	Writes the modified configuration to nonvolatile memory (NVRAM), permanently saving the settings.

Verifying the Enabling of the SNMP Agent

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

Step 1 Telnet to the target device:

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

where *xxx.xxx.xxx.xxx* represents the IP address of the target device.

Step 2 Establish the enable mode on the device:

```
Router# enable
```

Step 3 Display the running configuration on the device and examine the output for any displayed SNMP information:

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

Any “snmp-server” statements appearing in the output that takes the form shown above verifies that SNMP has been enabled on the specified device.

Configuration Examples

The following example shows how to enable an SNMP agent.

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

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

```
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 MPLS LDP MIB objects.

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

Command Reference

This section documents new or modified CLI commands applicable to this Cisco IOS release. All other CLI commands used with the MPLS LDP MIB feature are documented in the Cisco IOS Release 12.2 command reference publications.

snmp-server community

Use the **snmp-server community** global configuration command on your host network management station to configure read-only or read-write Simple Network Management Protocol (SNMP) community strings for the MPLS LDP MIB. Use the **no snmp-server community** command to change the community string to its default value.

snmp-server community *string* [**view** *view-name*] [**ro** | **rw**] [*number*]

no snmp-server community *string*

Syntax Description		
<i>string</i>		A community string consists of 1 to 32 alphanumeric characters and acts like a password, permitting access to SNMP functionality on LSRs in your network. Blank spaces are not allowed in the community string.
view <i>view-name</i>		(Optional). The name of a previously-defined view delineating the objects available to the SNMP community.
ro		(Optional). This default parameter configures read-only (RO) access to MPLS LDP MIBs on LSRs, thus limiting an authorized network management station (NMS) to retrieving MPLS LDP MIB objects only.
rw		(Optional). This parameter configures read-write (RW) access to MPLS LDP MIBs on LSRs. Accordingly, an authorized network management station can both retrieve and modify MPLS LDP MIB objects.
<i>number</i>		(Optional). This parameter is an integer from 1 to 99 specifying an access list of IP addresses that are allowed to use the community string to gain access to the SNMP v.1 agent.

Defaults The default value of the read/write parameter is read-only (**ro**). The default value of the read-only community string is public, and the default value of the read-write community string is private.

Command Modes Global configuration

Command History	Release	Modification
	10.0	This command was introduced.

Usage Guidelines The **no snmp-server** command disables both versions of SNMP (SNMPv1 and SNMPv2). The first **snmp-server** command that you enter enables both versions of SNMP.

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 assign the string *comaccess* to SNMPv1, allowing read-only access and specifying 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 *mgr* to SNMPv1, allowing read-write access to the objects in the restricted view:

```
Router(config)# snmp-server community mgr 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 both versions of SNMP:

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

Related Commands

Command	Description
snmp-server host	Specifies the recipient of an SNMP trap operation.
snmp-server enable-trap	Enables a router to send SNMP traps.

Glossary

inform request—A message sent by an SNMP agent to a network management station, console, or terminal to indicate that a significant event has occurred. SNMP inform requests are more reliable than traps (see below) because inform requests generally result in an acknowledgment message in the form of an SNMP response protocol data unit (PDU). If the intended recipient does not receive an inform request, it does not send a PDU. If the sender does not receive a PDU in response, it can resend the inform request. Thus, inform requests are more likely to reach their intended destination.

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.

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.

trap—A message sent by an SNMP agent to a network management station, console, or terminal to indicate that a significant event has occurred. Traps 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.

