

# MPLS Label Distribution Protocol MIB

This document describes the Simple Network Management Protocol (SNMP) agent support provided in Cisco software for the MPLS Label Distribution Protocol Management Information Base (MPLS LDP MIB).


- Restrictions for MPLS LDP MIB, on page 1
- Information About MPLS LDP MIB, on page 1
- How to Configure MPLS LDP MIB, on page 7
- Configuration Examples for MPLS LDP MIB, on page 12
- Additional References, on page 12
- Feature Information for MPLS LDP MIB, on page 13


# Restrictions for MPLS LDP MIB

The MPLS LDP MIB is limited to read-only (RO) permission for MIB objects, except for MIB object mplsLdpSessionUpDownTrapEnable, which is writable by the SNMP agent.

Setting this object to a value of true enables both the mplsLdpSessionUp and mplsLdpSessionDown notifications on the Label Switched Router (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 Events Generating MPLS LDP MIB Notifications section.

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.

# Information About MPLS LDP MIB

## MPLS LDP 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 a Hello adjacency, which represents the context within which label binding information is exchanged between the local LSR and its LDP peer. An LDP function 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.

# MPLS LDP MIB Overview

The MPLS LDP MIB has been implemented to enable standard, SNMP-based network management of the label switching features in Cisco software. 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 software 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 software.

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-08.txt*, which defines network management objects in a structured and standardized manner. This draft MIB is continually evolving 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 software require some minor translations between the MPLS LDP MIB objects and the internal data structures of Cisco software. Such translations are accomplished by the SNMP agent, which runs in the background on the NMS workstation as a low priority process.

The MPLS LDP MIB provides the following functions:

- The MPLS LDP MIB can generate and send event notification messages to signal changes in the status of LDP sessions.
- You can enable and disable event notification messages by using SNMP CLI commands.
- You can specify the name or the IP address of an NMS workstation where event notification messages are sent to serve network administrative and management purposes.
- You can store the configuration pertaining to an event notification message in nonvolatile 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** Because 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: ciscoExperiment 1.3.6.1.4.1.9.10 mplsLdpMIB 1.3.6.1.4.1.9.10.65 If the MPLS LDP MIB is assigned an IANA Experimental OID number, Cisco will deprecate all objects in the MIB under the Cisco Experimental OID and reposition the objects under the IANA Experimental OID.

# Benefits of Using MPLS LDP MIB

The MPLS LDP MIB provides the following benefits:

- 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
  - Determining the last active interface of an LDP session

# Description of MPLS LDP MIB Elements

The MPLS LDP MIB includes the following 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 nonlocal 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).

A Hello adjacency constitutes the working context between two LSRs in an MPLS network. The adjacency is used for the exchange of 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 tag, 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 between local entities and remote peers distribute 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-08.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 Configured 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 software.

The MPLS LDP MIB objects that announce 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.

The up and down notifications indicate the last active interface in the LDP session.

- 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 to 255; a value of 0 indicates that loop detection is off; any value other than 0 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.

We recommend 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 limits.

- 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 software and cannot be changed by 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 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 limit 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:

- • Nonoverlapping ATM VPI/VCI ranges (as noted above) or nonoverlapping Frame-Relay DLCI ranges between LSRs attempting to set up an LDP session

- Unsupported label distribution method
- Dissimilar protocol data unit (PDU) sizes
- Dissimilar LDP feature support

# How to Configure MPLS LDP MIB

## Enabling the SNMP Agent for the MPLS LDP MIB

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

### SUMMARY STEPS

1. **enable**
2. **show running-config**
3. **configure terminal**
4. **snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [*acl-number*]
5. **do copy running-config startup-config**
6. **exit**
7. **show running-config** [**interface** | **map-class**]

### DETAILED STEPS

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **show running-config**<br>**Example:**<br>`Router# show running-config` | Displays the running configuration to determine if an SNMP agent is already running.<br>• If no SNMP information is displayed, continue with the next step. If any SNMP information is displayed, you can modify the information or change it as needed. |
| **Step 3** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 4** | **snmp-server community** *string* [**view** *view-name*] [**ro** \| **rw**] [*acl-number*] | Sets up the community access string to permit access to the SNMP protocol. |

| | Command or Action | Purpose |
|---|---|---|
| | **Example:**<br>`Router(config)# snmp-server community comaccess ro` | • The *string* argument acts like a password and permits access to the SNMP protocol.<br>• The **view** *view-name* keyword and argument pair specifies the name of a previously defined view. The view defines the objects available to the community.<br>• The **ro** keyword specifies read-only access. Authorized management stations are only able to retrieve MIB objects.<br>• The **rw** keyword specifies read/write access. Authorized management stations are able to both retrieve and modify MIB objects.<br>• The *acl-number* argument is an integer from 1 to 99 that specifies an access list of IP addresses that are allowed to use the community string to gain access to the SNMP agent. |
| **Step 5** | **do copy running-config startup-config**<br>**Example:**<br>`Router(config)# do copy running-config startup-config` | Saves the modified configuration to nonvolatile memory (NVRAM) as the startup configuration file.<br>• The **do** command allows you to perform EXEC level commands in configuration mode. |
| **Step 6** | **exit**<br>**Example:**<br>`Router(config)# exit` | Returns to privileged EXEC mode. |
| **Step 7** | **show running-config** [**interface** \| **map-class**]<br>**Example:**<br>`Router# show running-config \| include smnp-server` | (Optional) Displays the configuration information currently on the router, the configuration for a specific interface, or map-class information.<br>• Use the **show running-config** command to check that the **snmp-server** statements appear in the output. |

# Configuring the Router to Send SNMP Traps

Perform this task to configure the router to send traps to a host.

The **snmp-server host** command specifies which hosts receive traps. The **snmp-server enable traps** command globally enables the trap production mechanism for the specified traps.

For a host to receive a trap, an **snmp-server host** command must be configured for that host, and, generally, the trap must be enabled globally through the **snmp-server enable traps** command.

**Note** Although you can set the *community-string* argument using the **snmp-server host** command by itself, we recommend that you define this string using the **snmp-server community** command prior to using the **snmp-server host** command.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server host** *host-addr* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] [**vrf** *vrf-name*]
4. **snmp-server enable traps mpls ldp** [**session-down**] [**session-up**] [**pv-limit**] [**threshold**]
5. **exit**

## DETAILED STEPS

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **snmp-server host** *host-addr* [**traps** \| **informs**] [**version** {**1** \| **2c** \| **3** [**auth** \| **noauth** \| **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] [**vrf** *vrf-name*]<br>**Example:**<br>`Router(config)# snmp-server host 172.20.2.160 traps comaccess mpls-ldp` | Specifies the recipient of an SNMP notification operation.<br>• The *host-addr* argument specifies the name or Internet address of the host (the targeted recipient).<br>• The **traps** keyword sends SNMP traps to this host. This is the default.<br>• The **informs** keyword sends SNMP informs to this host.<br>• The **version** keyword specifies the version of the SNMP used to send the traps. Version 3 is the most secure model, because it allows packet encryption with the **priv** keyword. If you use the **version** keyword, you must specify one of the following:<br>  • **1**--SNMPv1. This option is not available with informs.<br>  • **2c**--SNMPv2C. |

| | Command or Action | Purpose |
|---|---|---|
| | | • **3**--SNMPv3. The following three optional keywords can follow the **version 3** keyword (**auth**, **noauth**, **priv**).<br>• The *community-string* argument is a password-like community string sent with the notification operation.<br>• The **udp-port** *port* keyword argument pair names the UDP port of the host to use. The default is 162.<br>• The *notification-type* argument specifies the type of notification to be sent to the host. If no type is specified, all notifications are sent.<br>• The **vrf** *vrf-name* keyword argument pair specifies the VRF table that should be used to send SNMP notifications. |
| **Step 4** | **snmp-server enable traps mpls ldp** [**session-down**] [**session-up**] [**pv-limit**] [**threshold**]<br>**Example:**<br>`Router(config)# snmp-server enable traps mpls ldp session-down session-up` | Enables the router to send MPLS VPN- specific SNMP notifications (traps and informs).<br>• The **session-down** keyword controls (enables or disables) LDP session down notifications (mplsLdpSessionDown). This message is generated when an LDP session between the router and its adjacent LDP peer is terminated.<br>• The **session-up** keyword controls (enables or disables) LDP session up notifications (mplsLdpSessionUp). This notification is generated when the router establishes an LDP session with another LDP entity (an adjacent LDP peer in the network).<br>• The **pv-limit** keyword controls (enables or disables) path-vector (PV) limit notifications (mplsLdpPathVectorLimitMismatch). This notification is generated when the router establishes an LDP session with its adjacent peer LSR, but the two LSRs have dissimilar path vector limits.<br>• The **threshold** keyword controls (enables or disables) PV limit notifications (mplsLdpFailedInitSessionThresholdExceeded). This notification is generated after eight failed attempts to establish an LDP session between the router and an LDP peer. The failure can be the result of any type of incompatibility between the devices. |
| **Step 5** | **exit**<br>**Example:**<br>`Router(config)# exit` | (Optional) Exits to privileged EXEC mode. |

# Verifying the Status of the SNMP Agent

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

## SUMMARY STEPS

1. **enable**
2. **show running-config**
3. **exit**

## DETAILED STEPS

### Procedure

**Step 1** **enable**

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example:

**Example:**

```
Router> enable
Router#
```

**Step 2** **show running-config**

Use this command to display the running configuration on the host NMS and examine the output for SNMP information. For example:

**Example:**

```
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.

**Step 3** **exit**

Use this command to exit to user EXEC mode. For example:

**Example:**

```
Router# exit
Router>
```

# Configuration Examples for MPLS LDP MIB

## Enabling the SNMP Agent Examples

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

```
Router# configure 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 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 of the MPLS LDP MIB objects.

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

The following example shows how to enable the session up and session down LDP notifications:

```
Router(config)# snmp-server enable traps mpls ldp session-up
Router(config)# snmp-server enable traps mpls ldp session-down
```

# Additional References

### Related Documents

| Related Topic | Document Title |
|---|---|
| MPLS LDP configuration tasks | MPLS Label Distribution Protocol (LDP) |
| MPLS LDP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples | *Cisco IOS Multiprotocol Label Switching Command Reference* |
| SNMP commands | *Cisco IOS Network Management Command Reference* |
| SNMP configuration | "Configuring SNMP Support" in the *Network Management Configuration Guide* |

### Standards

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

### MIBs

| MIBs | MIBs Link |
|---|---|
| • *MPLS LDP MIB* | To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL:<br>http://www.cisco.com/go/mibs |

### RFCs

| RFCs | Title |
|---|---|
| RFC 3036 | *LDP Specification* |
| RFC 3037 | *LDP Applicability* |

### Technical Assistance

| Description | Link |
|---|---|
| The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.<br>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.<br>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password. | http://www.cisco.com/techsupport |

# Feature Information for MPLS LDP MIB

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

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

*Table 1: Feature Information for MPLS LDP MIB*

| Feature Name | Releases | Feature Information |
|---|---|---|
| MPLS LDP MIB | 12.0(11)ST<br>12.2(2)T<br>12.0(21)ST<br>12.2(13)T<br>12.0(30)S<br>12.2(27)SBC<br>12.2(28)SB<br>12.2(33)SRA<br>12.2(33)SXH<br>Cisco IOS XE Release 2.1 | The MPLS LDP MIB has been implemented to enable standard, SNMP-based network management of the label switching features in Cisco software.<br>In Cisco IOS Release 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.<br>In Cisco IOS Release 12.2(2)T, this feature was integrated to provide SNMP agent support when using the MPLS LDP MIB on Cisco 7200 and Cisco 7500 series routers.<br>In Cisco IOS Release 12.0(21)ST, the **snmp-server enable traps mpls ldp** command was introduced.<br>The **snmp-server enable traps mpls ldp** command was integrated into Cisco IOS Release 12.2(13)T.<br>This feature was integrated into Cisco IOS Release 12.0(30)S.<br>This feature was integrated into Cisco IOS Release 12.2(27)SBC.<br>This feature was integrated into Cisco IOS Release 12.2(28)SB.<br>This feature was integrated into Cisco IOS Release 12.2(33)SRA.<br>This feature was integrated into Cisco IOS Release 12.2(33)SXH.<br>This feature was integrated into Cisco IOS XE Release 2.1 and implemented on the Cisco ASR 1000 Series Aggregation Services Routers.<br>The following commands were introduced or modified: **snmp-server enable traps mpls ldp** |
| MPLS LDP—MIB Notifications | Cisco IOS XE Release 2.1 | This feature provides SNMP traps for critical MPLS LDP events.<br>This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.<br>The following commands were introduced or modified: **snmp-server enable traps mpls ldp** |