



MPLS Embedded Management and MIBs Configuration Guide, Cisco IOS Release 12.4

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

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

© 2011 Cisco Systems, Inc. All rights reserved.



CONTENTS

MPLS Enhancements to Interfaces MIB 1

- Finding Feature Information 1
- Prerequisites for MPLS Enhancements to Interfaces MIB 1
- Restrictions for MPLS Enhancements to Interfaces MIB 1
- Information About MPLS Enhancements to Interfaces MIB 2
 - Feature Design of the MPLS Enhancements to Interfaces MIB 3
 - ifStackTable Objects 3
 - ifRcvAddressTable Objects 4
 - Interfaces MIB Scalar Objects 5
 - Stacking Relationships for MPLS Layer Interfaces 5
 - Stacking Relationships for Traffic Engineering Tunnels 6
 - MPLS Label Switching Router MIB Enhancements 7
 - Benefits of the MPLS Enhancements to Interfaces MIB 8
- How to Configure MPLS Enhancements to Interfaces MIB 8
 - Enabling the SNMP Agent 8
- Configuration Examples for the MPLS Enhancements to Interfaces MIB 10
 - MPLS Enhancements to Interfaces MIB: Examples 10
- Additional References 10
- Feature Information for MPLS Enhancements to Interfaces MIB 11
- Glossary 12

MPLS Label Switching Router MIB 15

- Finding Feature Information 16
- Information About MPLS Label Switching Router MIB 16
 - MPLS-LSR-MIB Elements 17
 - MPLS-LSR-MIB Tables 17
 - Information from Scalar Objects 21
 - Linking Table Elements 21
 - Interface Configuration Table and Interface MIB Links 22
 - Using the MPLS-LSR-MIB 24

MPLS-LSR-MIB Structure	24
CLI Commands and the MPLS-LSR-MIB	25
CLI Command Output	26
MPLS-LSR-MIB Output	26
Benefits	26
How to Configure the MPLS LSR MIB	27
Prerequisites	27
Enabling the SNMP Agent	27
Verifying That the SNMP Agent Has Been Enabled	28
Configuration Examples for the MPLS LSR MIB	29
Additional References	30
Command Reference	31
Glossary	31
MPLS Label Distribution Protocol MIB Version 8 Upgrade	35
Finding Feature Information	35
Prerequisites for MPLS LDP MIB Version 8 Upgrade	35
Restrictions for MPLS LDP MIB Version 8 Upgrade	35
Information About MPLS LDP MIB Version 8 Upgrade	36
Feature Design of MPLS LDP MIB Version 8 Upgrade	36
Enhancements in Version 8 of the MPLS LDP MIB	38
Benefits of MPLS LDP MIB Version 8 Upgrade	38
Description of MPLS LDP MIB Elements for MPLS LDP MIB Version 8 Upgrade	38
LDP Entities	39
LDP Sessions and Peers	40
LDP Hello Adjacencies	41
Events Generating MPLS LDP MIB Notifications in MPLS LDP MIB Version 8 Upgrade	42
MIB Tables in MPLS LDP MIB Version 8 Upgrade	43
mplsLdpEntityTable	44
mplsLdpEntityConfGenLRTable	47
mplsLdpEntityAtmParmsTable	48
mplsLdpEntityConfAtmLRTable	49
mplsLdpEntityStatsTable	49
mplsLdpPeerTable	51
mplsLdpHelloAdjacencyTable	51
mplsLdpSessionTable	52

mplsLdpAtmSesTable	53
mplsLdpSesStatsTable	53
VPN Contexts in MPLS LDP MIB Version 8 Upgrade	54
SNMP Context	54
VPN Aware LDP MIB Sessions	54
VPN Aware LDP MIB Notifications	56
How to Configure MPLS LDP MIB Version 8 Upgrade	58
Enabling the SNMP Agent	58
Enabling Distributed Cisco Express Forwarding	60
Enabling MPLS Globally	61
Enabling LDP Globally	62
Enabling MPLS on an Interface	62
Enabling LDP on an Interface	63
Configuring a VPN Aware LDP MIB	64
Configuring SNMP Support for a VPN	65
Configuring an SNMP Context for a VPN	66
SNMP Context	66
VPN Route Distinguishers	66
Associating an SNMP VPN Context with SNMPv1 or SNMPv2	68
Verifying MPLS LDP MIB Version 8 Upgrade	70
Configuration Examples for MPLS LDP MIB Version 8 Upgrade	71
MPLS LDP MIB Version 8 Upgrade Examples	71
Configuring a VPN Aware SNMP Context for SNMPv1 or SNMPv2 Example	71
Additional References	72
Feature Information for MPLS LDP MIB Version 8 Upgrade	73
Glossary	75
MPLS Traffic Engineering MIB	79
Finding Feature Information	79
Restrictions for the MPLS Traffic Engineering MIB	79
Information About the MPLS Traffic Engineering MIB	80
MPLS Traffic Engineering MIB Cisco Implementation	80
MPLS Traffic Engineering Overview	80
Capabilities Supported by the MPLS Traffic Engineering MIB	80
Notification Generation Events	81
Notification Implementation	81

Benefits of the MPLS Traffic Engineering MIB	82
MPLS Traffic Engineering MIB Layer Structure	82
Features and Technologies Related to MPLS Traffic Engineering MIB	82
Supported Objects in the MPLS Traffic Engineering MIB	82
CLI Access to MPLS Traffic Engineering MIB Information	87
Retrieving Information from the MPLS Traffic Engineering MIB	87
How to Configure the MPLS Traffic Engineering MIB	88
Enabling the SNMP Agent to Help Manage Various MPLS TE Tunnel Characteristics of Tunnels on the Local Router	88
Verifying the Status of the SNMP Agent	89
Examples	90
Configuration Examples for the MPLS Traffic Engineering MIB	90
Example Enabling the SNMP Agent to Help Manage MPLS TE Characteristics of Tunnels on the Local Router	91
Additional References	91
Feature Information for the MPLS Traffic Engineering MIB	92
Glossary	93
Monitoring MPLS VPNs with MIBs	97
Finding Feature Information	97
Prerequisites for PPVPN MPLS VPN MIB	97
Restrictions for PPVPN MPLS VPN MIB	98
Information About PPVPN MPLS VPN MIB	98
MPLS VPN Overview	98
PPVPN MPLS VPN MIB Overview	98
PPVPN MPLS VPN MIB and the IETF	99
Capabilities Supported by PPVPN-MPLS-VPN MIB	99
Functional Structure of the PPVPN-MPLS-VPN MIB	99
Supported Objects in PPVPN-MPLS-VPN MIB	100
Scalar Objects	101
MIB Tables	101
mplsVpnVrfTable	101
mplsVpnInterfaceConfTable	104
mplsVpnVrfRouteTargetTable	105
mplsVpnVrfBgpNbrAddrTable	107
mplsVpnVrfSecTable	108

mplsVpnVrfPerfTable	108
mplsVpnVrfRouteTable	109
Notifications	112
PPVPN-MPLS-VPN MIB Notification Events	112
CISCO-IETF-PPVPN-MPLS-VPN MIB Notification Events	113
Notification Specification	114
Monitoring the PPVPN-MPLS-VPN MIB Notifications	114
MIB Objects Not Supported	114
How to Configure PPVPN MPLS VPN MIB	115
Configuring the SNMP Community	115
Configuring the Router to Send SNMP Traps	116
Configuring Threshold Values for MPLS VPN--SNMP Notifications	119
Configuration Examples for PPVPN MPLS VPN MIB	121
Configuring the SNMP Community Examples	121
Configuring the Router to Send SNMP Traps Example	121
Configuring Threshold Values for MPLS VPN--SNMP Notifications Examples	121
Additional References	122
Feature Information for PPVPN MPLS VPN MIB	123



MPLS Enhancements to Interfaces MIB

This document describes the Multiprotocol Label Switching (MPLS) enhancements to the existing Interfaces MIB (RFC 2233) to support an MPLS layer. This layer provides counters and statistics specifically for MPLS.

- [Finding Feature Information, page 1](#)
- [Prerequisites for MPLS Enhancements to Interfaces MIB, page 1](#)
- [Restrictions for MPLS Enhancements to Interfaces MIB, page 1](#)
- [Information About MPLS Enhancements to Interfaces MIB, page 2](#)
- [How to Configure MPLS Enhancements to Interfaces MIB, page 8](#)
- [Configuration Examples for the MPLS Enhancements to Interfaces MIB, page 10](#)
- [Additional References, page 10](#)
- [Feature Information for MPLS Enhancements to Interfaces MIB, page 11](#)
- [Glossary, page 12](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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.

Prerequisites for MPLS Enhancements to Interfaces MIB

- Simple Network Management Protocol (SNMP) must be installed and enabled on the label switching routers (LSRs)
- MPLS must be enabled on the LSRs
- MPLS IP must be enabled on an interface or an MPLS traffic engineering (TE) tunnel enabled on an interface

Restrictions for MPLS Enhancements to Interfaces MIB

- Link up and link down traps for the MPLS layer are not supported in this release.

- Write capability using the SNMP SET command is not supported for the MPLS layer in this release.
- Some counters, including discard and multicast, increment on the underlying physical layer; therefore, they equal 0 because they never reach the MPLS layer.
- The high-capacity counters for the MPLS layer interfaces of the Interfaces MIB contain 64 bits of counter data. In previous versions, the high capacity counters displayed 32 bits of counter data.

The following MIB objects are affected:

- - ifHCInOctets
 - ifHCOctets
 - ifHCInUcastPkts
 - ifHCOctetsUcastPkts

When the 64-bit values are less than the value of 232, the 32-bit and 64-bit values are identical.

After the counter increases to more than 232, the counters are different; the 64-bit value is computed by the following formula:

$$X * (232) + Y$$

where:

- - X is the number of times the 32-bit counter has rolled.
 - Y is the residual value of the counter after the roll occurred. The Y value equals the 32-bit value.

When the high-capacity counter values are compared to their 32-bit values, there is a period of time that the counter values are not equal. The 64-bit values lag the 32-bit values when the counters poll the 32-bit hardware counters and computing the correct counter value. During the polling and computation interval, the following high-capacity counter values counters might be inconsistent:

- - ifInOctets
 - ifOutOctets
 - ifInUcastPkts
 - ifOutUcastPkts

The inconsistent values can occur if traffic is constantly flowing over an interface and a MIB walk is performed. The 32-bit value is correct at that moment. The 64-bit value lags slightly, because of the polling computations needed to generate it. Once traffic stops flowing over the interface, and a polling period has passed, the two counters are identical and correct.

The lag time depends on the following factors:

- - The polling interval used by the Interfaces MIB. The less time the polling interval takes, the more accurate the value is.
 - The size of the Interfaces MIB. A large MIB takes a long time to walk and might affect the values found at that instant.
 - The number of computations needed to generate the 64-bit value. The number of MPLS-enabled interfaces increases the number of 64-bit counter values that need to be computed.

Information About MPLS Enhancements to Interfaces MIB

- [Feature Design of the MPLS Enhancements to Interfaces MIB, page 3](#)
- [Interfaces MIB Scalar Objects, page 5](#)
- [Stacking Relationships for MPLS Layer Interfaces, page 5](#)

- [Stacking Relationships for Traffic Engineering Tunnels, page 6](#)
- [MPLS Label Switching Router MIB Enhancements, page 7](#)
- [Benefits of the MPLS Enhancements to Interfaces MIB, page 8](#)

Feature Design of the MPLS Enhancements to Interfaces MIB

The Interfaces MIB (IF MIB) provides an SNMP-based method for managing interfaces. Each entry in the IF MIB establishes indexing, statistics, and stacking relationships among underlying physical interfaces, subinterfaces, and Layer 2 protocols that exist within Cisco software.

The enhancements add an MPLS layer to the IF MIB as a Layer 2 protocol to provide statistics for traffic encapsulated as MPLS on an interface. In this structure, MPLS-specific data such as MPLS-encapsulated traffic counters and the MPLS maximum transmission unit (MTU) resides on top of the underlying physical or virtual interface to allow separation from non-MPLS data.

The enhancements also allow you to display indexing, statistics, and stacking relationships using the ifStackTable. MPLS layer interfaces are stacked above the underlying physical or virtual interface that is actually forwarding the MPLS traffic. MPLS traffic engineering tunnels are then stacked above those MPLS layers.

The IF MIB supports several types of interfaces. A virtual interface that provides protocol statistics for MPLS-encapsulated traffic has been added. This interface is stacked above real Cisco interfaces or subinterfaces, such as Fast Ethernet (fe0/1/0) or ATM (at1/1.1).

Cisco software creates a corresponding MPLS layer above each interface capable of supporting MPLS when the MPLS encapsulation is enabled by issuing the **mpls ip** command in interface configuration mode.

You can also create the interface layer if you enable MPLS TE by using the **mpls traffic-eng tunnels** command in interface configuration mode.



Note

You must also issue these commands in global configuration mode for MPLS IP or MPLS TE to be enabled.

An IF MIB entry is created when you enable either MPLS IP or MPLS TE tunnels on an interface; the entry is removed when you disable both MPLS IP and MPLS TE.

- [ifStackTable Objects, page 3](#)
- [ifRcvAddressTable Objects, page 4](#)

ifStackTable Objects

The table below defines the ifStackTable objects.

Table 1 *ifStackTable Objects and Definitions*

Object	Definition
ifStackHigherLayer	The value of ifIndex corresponding to the higher sublayer of the relationship; that is, the sublayer that runs on top of the sublayer identified by the corresponding instance of the ifStackLowerLayer. Note Index objects are not accessible in a MIB walk. This value is part of the object identifier (OID) for every object in the ifStackTable.
ifStackLowerLayer	The value of ifIndex corresponding to the lower sublayer of the relationship; that is, the sublayer that runs below the sublayer identified by the corresponding instance of the ifStackHigherLayer. Note Index objects are not accessible in a MIB walk. This value is part of the OID for every object in the ifStackTable.
ifStackStatus	Used to create and delete rows in the ifStackTable; status is always active(1) for MPLS.

ifRcvAddressTable Objects

The table below defines the ifRcvAddressTable objects.



Note

Entries for the MPLS layer do not appear in the ifRcvAddressTable.

Table 2 *ifRcvAddressTable Objects and Descriptions*

Object	Definition
ifRcvAddressAddress	An address for which the system accepts packets and frames on this entry's interface. Note Index objects are not accessible in a MIB walk. This value is part of the OID for every object in the ifRcvAddressTable.
ifRcvAddressStatus	Used to create and delete rows in the ifRcvAddressTable.
ifRcvAddressType	Type of storage used for each entry in the ifRcvAddressTable.

Interfaces MIB Scalar Objects

The IF MIB supports the following scalar objects:

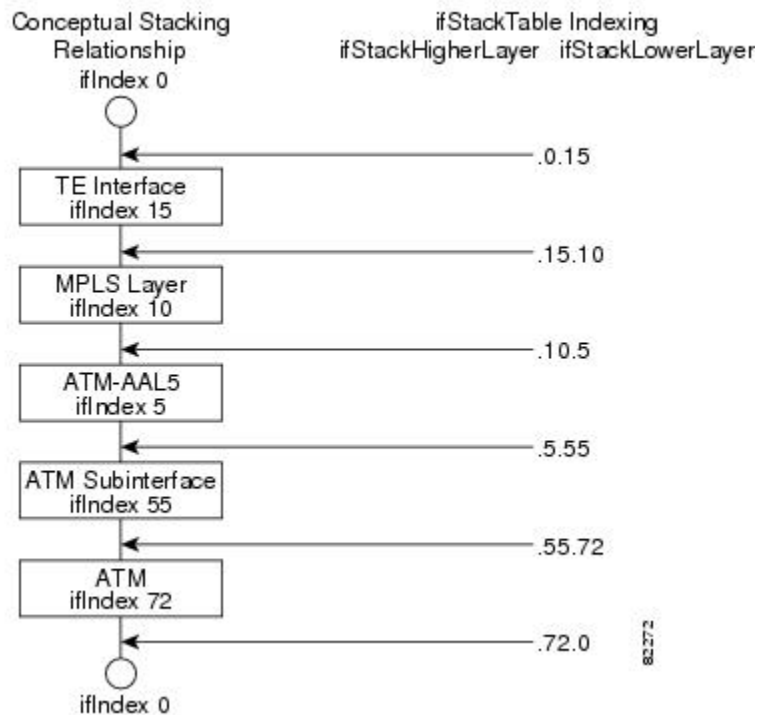
- ifStackLastChange--The value of sysUpTime at the time of the last change of the entire interface stack. A change of the interface stack is defined to be any creation, deletion, or change in value of any instance of ifStackStatus. If the interface stack has been unchanged since the last reinitialization of the local network management subsystem, then this object contains a zero value.
- ifTableLastChange--The value of sysUpTime at the time of the last creation or deletion of an entry in the ifTable. If the number of entries has been unchanged since the last reinitialization of the local network management subsystem, then this object contains a zero value.

Stacking Relationships for MPLS Layer Interfaces

The ifStackTable within the IF MIB provides a conceptual stacking relationship between the interfaces and subinterfaces represented as entries in the ifTable.

The ifStackTable is indexed like a linked list. Each entry shows a relationship between two interfaces providing the ifIndexes of the upper and the lower interface. The entries chain together to show the entire stacking relationship. Each entry links with one another until the stack terminates with an ifIndex of 0 at the highest and lowest ends of the stack. For example, in the figure below, the indexes .10.5 show that ifIndex 10 is stacked upon ifIndex 5. There are 0 entries at the highest and lowest ends of the stack; in the figure, the indexes .0.15 and .72.0 are the highest and lowest ends of the stack, respectively.

Figure 1 Sample ATM Stacking Relationship in the ifStackTable



The table below describes the indexing of the ifStackTable for the layer relationships shown in the figure above.

**Note**

The order of the entries in the table may not be the same as that seen in the MIB walk, which has to follow SNMP ordering rules.

Table 3 *Layer Relationships*

Layer Relationship (in Descending Order)	ifStackHigherLayer/ifStackLowerLayer
TE interface as top layer	.0.15
TE interface stacked upon MPLS layer	.15.10
MPLS layer stacked upon ATM-AAL5	.10.5
ATM-AAL5 layer stacked upon ATM subinterface	.5.55
ATM subinterface stacked upon ATM	.55.72
ATM as bottom layer	.72.0

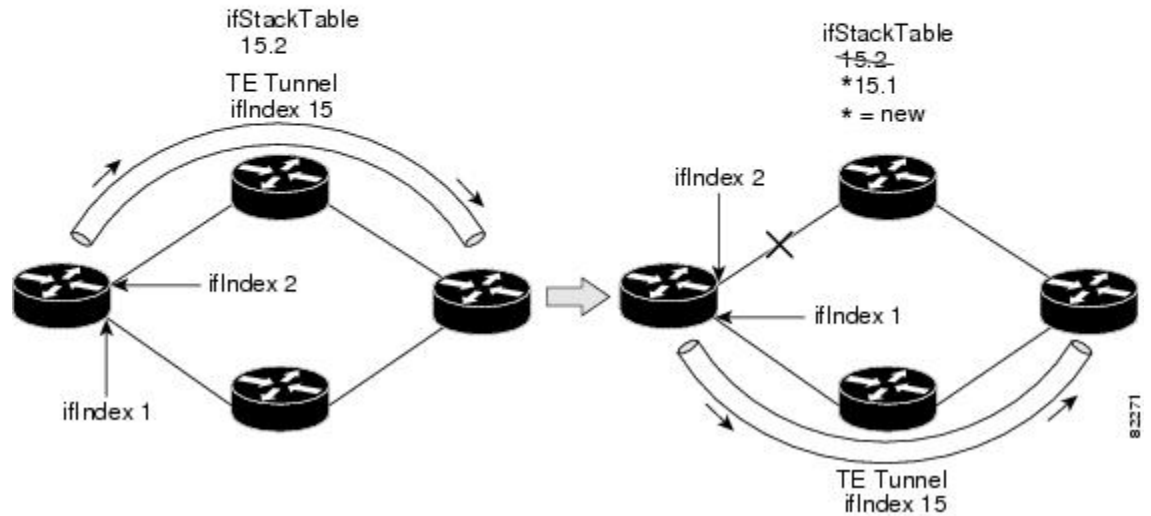
Stacking Relationships for Traffic Engineering Tunnels

MPLS TE tunnels are represented in Cisco software and the IF MIB as virtual interfaces. When properly signaled, TE tunnels pass traffic through MPLS over a physical interface. This process dictates that a TE tunnel is to be stacked on an MPLS layer that is stacked on an underlying interface.

TE tunnels can also change paths in response to different error or network conditions. These changes are instigated by using the RSVP-TE signaling protocol. When a change occurs, a tunnel can switch to a different MPLS interface. If no signaling path exists, no paths will be chosen and thus no MPLS interface will be used.

Because a TE tunnel is represented as an IF MIB ifTable entry, the ifStackTable also contains an entry corresponding to the TE tunnel. If the TE tunnel is successfully signaled, the ifStackTable also contains a link between the tunnel interface and one MPLS interface. Note that because it is possible for a TE tunnel to not have a corresponding signaled path, it is thus possible for a TE tunnel's ifStackTable entry to not have a corresponding lower layer. In this case, the lower layer variable contains the value of 0.

The figure below shows a TE tunnel before (left) and after (right) being rerouted and the effect on the ifStackTable. When ifIndex 2 fails, the TE tunnel is rerouted through ifIndex1, the 15.2 entry is removed from the ifStackTable, and the 15.1 entry is added.



MPLS Label Switching Router MIB Enhancements

All of the ifIndex references in the MPLS-LSR-MIB tables have changed from the ifIndex of the underlying physical or virtual interface to the ifIndex of the MPLS layer.

The table below shows the specific changes.

Table 4 *MPLS-LSR-MIB ifIndex Objects Enhanced*

Table	ifIndex
MPLS interface configuration table (mplsInterfaceConfTable)	mplsInterfaceConfIndex
MPLS in-segment table (mplsInSegmentTable)	mplsInSegmentIfIndex
MPLS cross-connect table (mplsXCTable)	mplsInSegmentIfIndex
MPLS out-segment table (mplsOutSegmentTable)	mplsOutSegmentIfIndex

The following objects from the mplsInterfaceConfTable are affected:

- mplsInterfaceOutPackets--Count only MPLS-encapsulated out packets
- mplsInterfaceInPackets--Count only MPLS-encapsulated in packets

Benefits of the MPLS Enhancements to Interfaces MIB

Improved Accounting Capability

By viewing the MPLS layer, you get MPLS-encapsulated traffic counters that do not include non-MPLS encapsulated traffic (for example, IP packets). Therefore, the counters are more useful for MPLS-related statistics.

TE Tunnel Interfaces

For TE tunnel interfaces, the stacking relationship reflects the current underlying MPLS interface that is in use and dynamically changes as TE tunnels reoptimize and reroute.

MPLS-Specific Information

The MPLS layer shows MPLS-specific information including the following:

- If MPLS is enabled
- MPLS counters
- MPLS MTU
- MPLS operational status

How to Configure MPLS Enhancements to Interfaces MIB

- [Enabling the SNMP Agent, page 8](#)

Enabling the SNMP Agent

SUMMARY STEPS

1. `enable`
2. `show running-config`
3. `configure terminal`
4. `snmp-server community string [view view-name] [ro number]`
5. `end`
6. `write memory`
7. `show running-config`

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 enable</p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
<p>Step 2 show running-config</p> <p>Example:</p> <pre>Router# show running-config</pre>	<p>Displays the running configuration of the router so that you can determine if an SNMP agent is already running on the device.</p> <p>If no SNMP information is displayed, continue with the next step.</p> <p>If any SNMP information is displayed, you can modify the information or change it as desired.</p>
<p>Step 3 configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 4 snmp-server community string [view view-name] [ro number]</p> <p>Example:</p> <pre>Router(config)# snmp-server community public ro</pre>	<p>Configures read-only (ro) community strings for the MPLS Label Distribution Protocol (LDP) MIB.</p> <ul style="list-style-type: none"> The <i>string</i> argument functions like a password, permitting access to SNMP functionality on label switch routers (LSRs) in an MPLS network. The optional ro keyword configures read-only (ro) access to the objects in the MPLS LDP MIB.
<p>Step 5 end</p> <p>Example:</p> <pre>Router(config)# end</pre>	<p>Exits to privileged EXEC mode.</p>
<p>Step 6 write memory</p> <p>Example:</p> <pre>Router# write memory</pre>	<p>Writes the modified SNMP configuration into NVRAM of the router, permanently saving the SNMP settings.</p>
<p>Step 7 show running-config</p> <p>Example:</p> <pre>Router# show running-config</pre>	<p>Displays the running configuration of the router so that you can determine if an SNMP agent is already running on the device.</p> <p>If you see any snmp-server statements, SNMP has been enabled on the router.</p> <p>If any SNMP information is displayed, you can modify the information or change it as desired.</p>

Configuration Examples for the MPLS Enhancements to Interfaces MIB

- [MPLS Enhancements to Interfaces MIB: Examples, page 10](#)

MPLS Enhancements to Interfaces MIB: Examples

The following example shows how to enable an SNMP agent:

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

In the following example, SNMPv1 and SNMPv2C are enabled. The configuration permits any SNMP manager to access all objects with read-only permissions using the community string *public*.

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

In the following example, read-only access is allowed for all objects to members of access list 4 that specify the comaccess community string. No other SNMP managers have access to any objects.

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

Additional References

Related Documents

Related Topic	Document Title
SNMP commands	<i>Cisco IOS Network Management Command Reference</i>
SNMP configuration	“Configuring SNMP Support” in the <i>Network Management Configuration Guide</i> .
A description of SNMP agent support for the MPLS Traffic Engineering MIB (MPLS TE MIB)	MPLS Traffic Engineering (TE) MIB

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
<i>Interfaces Group MIB (IF MIB)</i>	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 1156	<i>Management Information Base for Network Management of TCP/IP-based internets</i>
RFC 1157	<i>A Simple Network Management Protocol (SNMP)</i>
RFC 1213	<i>Management Information Base for Network Management of TCP/IP-based internets: MIB-II</i>
RFC 1229	<i>Extensions to the Generic-Interface MIB</i>
RFC 2233	<i>Interfaces MIB</i>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>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.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Feature Information for MPLS Enhancements to Interfaces 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 5 *Feature Information for MPLS Enhancements to Interfaces MIB*

Feature Name	Releases	Feature Information
MPLS Enhancements to Interfaces MIB	12.0(23)S 12.3(8)T 12.2(33)SRA 12.2(33)SXH 12.2(33)SB Cisco IOS XE Release 2.1	This document describes the Multiprotocol Label Switching (MPLS) enhancements to the existing Interfaces MIB (RFC 2233) to support an MPLS layer. This layer provides counters and statistics specifically for MPLS. In Cisco IOS Release 12.0(23)S, this feature was introduced. This feature was integrated into Cisco IOS Release 12.3(8)T. This feature was integrated into Cisco IOS Release 12.2(33)SRA. This feature was integrated into Cisco IOS Release 12.2(33)SXH. This feature was integrated into Cisco IOS Release 12.2(33)SB. In Cisco IOS XE Release 2.1, this feature was implemented on the Cisco ASR 1000 Series Aggregation Services Routers. The following command was introduced or modified: snmp-server community .

Glossary

ATM -- Asynchronous Transfer Mode. The international standard for cell relay in which multiple service types (such as voice, video, or data) are conveyed in fixed-length (53-byte) cells. Fixed-length cells allow cell processing to occur in hardware, thereby reducing transit delays. ATM is designed to take advantage of high-speed transmission media, such as E3, SONET, and T3.

ATM-AAL5 --ATM adaptation layer 5. One of four AALs recommended by the ITU-T. AAL5 supports connection-oriented variable bit rate (VBR) services and is used predominantly for the transfer of classical IP over ATM and LAN emulation (LANE) traffic. AAL5 uses simple and efficient AAL (SEAL) and is the least complex of the current AAL recommendations. It offers low bandwidth overhead and simpler processing requirements in exchange for reduced bandwidth capacity and error-recovery capability.

encapsulation -- Wrapping of data in a particular protocol header. For example, Ethernet data is wrapped in a specific Ethernet header before network transit. Also, when bridging dissimilar networks, the entire

frame from one network is simply placed in the header used by the data link layer protocol of the other network.

IETF --Internet Engineering Task Force. A task force (consisting of more than 80 working groups) that is developing standards for the Internet and the IP suite of protocols.

interface --The boundary between adjacent layers of the ISO model.

label --A short, fixed-length identifier that is used to determine the forwarding of a packet.

label switching--A term used to describe the forwarding of IP (or other network layer) packets using a label swapping algorithm based on network layer routing algorithms. The forwarding of these packets uses the exact match algorithm and rewrites the label.

LSR --label switching router. A device that forwards Multiprotocol Label Switching (MPLS) packets based on the value of a fixed-length label encapsulated in each packet.

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

MPLS --Multiprotocol Label Switching. A method for forwarding packets (frames) through a network. It enables routers at the edge of a network to apply labels to packets (frames). ATM switches or existing routers in the network core can switch packets according to the labels with minimal lookup overhead.

MPLS interface--An interface on which Multiprotocol Label Switching (MPLS) traffic is enabled.

MTU --maximum transmission unit. Maximum packet size, in bytes, that a particular interface can handle.

NMS --network management system. System responsible for managing at least part of a network. An NMS is generally a reasonably powerful and well-equipped computer, such as an engineering workstation. NMSs communicate with agents to help keep track of network statistics and resources.

OID --object identifier. Values are defined in specific MIB modules. The Event MIB allows you or an NMS to watch over specified objects and to set event triggers based on existence, threshold, and Boolean tests. An event occurs when a trigger is fired; this means that a specified test on an object returns a value of true. To create a trigger, you or a network management system (NMS) configures a trigger entry in the `mteTriggerTable` of the Event MIB. This trigger entry specifies the OID of the object to be watched. For each trigger entry type, corresponding tables (existence, threshold, and Boolean tables) are populated with the information required for carrying out the test. The MIB can be configured so that when triggers are activated (fired) either a Simple Network Management Protocol (SNMP) Set is performed, a notification is sent out to the interested host, or both.

SNMP --Simple Network Management Protocol. A management protocol used almost exclusively in TCP/IP networks. SNMP provides a means for monitoring and controlling network devices, and for managing configurations, statistics collection, performance, and security.

traffic engineering tunnel--A label-switched tunnel that is used for traffic engineering. Such a tunnel is set up through means other than normal Layer 3 routing; it is used to direct traffic over a path different from the one that Layer 3 routing could cause the tunnel to take.

trap --A message sent by a Simple Network Management Protocol (SNMP) agent to a network management station, console, or terminal, indicating that a significant event occurred. Traps are less reliable than notification requests, because the receiver does not send an acknowledgment when it receives a trap. The sender cannot determine if the trap was received.

tunnel --A secure communication path between two peers, such as routers.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

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



MPLS Label Switching Router MIB

The MPLS Label Switching Router MIB (MPLS-LSR-MIB) allows you to use the Simple Network Management Protocol (SNMP) to remotely monitor a label switch router (LSR) that is using the Multiprotocol Label Switching (MPLS) technology.

Scalability enhancements provided in the Cisco IOS 12.0(28)S release reduce the size of any MIB walk and improve the usability of the MPLS-LSR-MIB.



Note

In Cisco IOS Release 12.2(33)SRB and Cisco IOS Release 12.2(33)SB, this MIB has been deprecated and replaced by MPLS-LSR-STD-MIB (RFC 3813). In those two releases and in later images, the entire MIB can be referenced by the name `mplsLsrMIB` for purposes of the SNMP server excluded/included command. If other MIB object names need to be referenced on the router, they must be referenced by `MPLS-LSR-MIB::<table_entry_name>`.

Feature History for MPLS Label Switching Router MIB

Release	Modification
12.0(14)ST	This feature was introduced on Cisco IOS Release 12.0(14)ST
12.2(2)T	This feature was integrated into Cisco IOS Release 12.2(2)T.
12.0(22)S	This feature was implemented on the Cisco 12000 series routers and integrated into Cisco IOS Release 12.0(22)S.
12.2(14)S	This feature was integrated into Cisco IOS Release 12.2(14)S and implemented on Cisco 7200 and Cisco 7500 series routers.
12.2(25)S	This feature was updated to work in the MPLS High Availability environment with the Cisco 7500 series routers.
12.0(28)S	This feature was updated to include scalability enhancements in Cisco IOS Release 12.0(28)S.

Release	Modification
12.2(33)SRB	This MIB has been deprecated and replaced by MPLS-LSR-STD-MIB (RFC 3813).
12.2(33)SB	This MIB has been deprecated and replaced by MPLS-LSR-STD-MIB (RFC 3813).

- [Finding Feature Information, page 16](#)
- [Information About MPLS Label Switching Router MIB, page 16](#)
- [How to Configure the MPLS LSR MIB, page 27](#)
- [Configuration Examples for the MPLS LSR MIB, page 29](#)
- [Additional References, page 30](#)
- [Command Reference, page 31](#)
- [Glossary, page 31](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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.

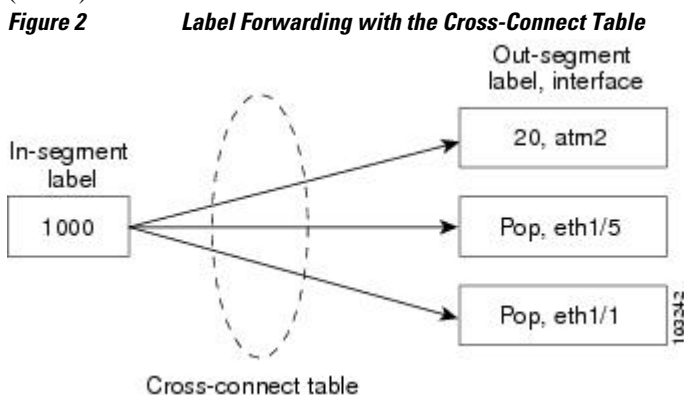
Information About MPLS Label Switching Router MIB

The MPLS-LSR-MIB contains managed objects that support the retrieval of label switching information from a router. The MIB is based on Revision 05 of the IETF MPLS-LSR-MIB. The MPLS-LSR-MIB mirrors a portion of the Cisco MPLS subsystem; specifically, it mirrors the Label Forwarding Information Base (LFIB). This implementation enables a network administrator to get information on the status, character, and performance of the following:

- MPLS-capable interfaces on the LSR
- Incoming MPLS segments (labels) at an LSR and their associated parameters
- Outgoing segments (labels) at an LSR and their associated parameters

In addition, the network administrator can retrieve the status of cross-connect table entries that associate MPLS segments with each other.

The figure below shows the association of the cross-connect table with incoming and outgoing segments (labels).



**Note**

The out-segment table does not display “no label” entries. Labels that are displayed as “POP” are the special MPLS label 3.

The notation used in the MPLS-LSR-MIB follows the conventions defined in Abstract Syntax Notation One (ASN.1). ASN.1 defines an Open System Interconnection (OSI) language used to describe data types apart from particular computer structures and presentation techniques. Each object in the MIB incorporates a DESCRIPTION field that includes an explanation of the object’s meaning and usage, which, together with the other characteristics of the object (SYNTAX, MAX-ACCESS, and INDEX) provides sufficient information for management application development, as well as for documentation and testing.

The MPLS-LSR-MIB represents an ASN.1 notation reflecting an idealized MPLS LSR.

A network administrator can access the entries (objects) in the MPLS-LSR-MIB by means of any SNMP-based network management system (NMS). The network administrator can retrieve information in the MPLS-LSR-MIB using standard SNMP **get** and **getNext** operations.

Typically, SNMP runs as a low-priority process. The response time for the MPLS-LSR-MIB is expected to be similar to that for other MIBs. The size and structure of the MIB and other MIBs in the system influence response time when you retrieve information from the management database. Traffic through the LSR also affects SNMP performance. The busier the switch is with forwarding activities, the greater the possibility of lower SNMP performance.

- [MPLS-LSR-MIB Elements, page 17](#)
- [Linking Table Elements, page 21](#)
- [Interface Configuration Table and Interface MIB Links, page 22](#)
- [Using the MPLS-LSR-MIB, page 24](#)
- [Benefits, page 26](#)

MPLS-LSR-MIB Elements

The top-level components of the MPLS-LSR-MIB consist of

- Tables and scalars (mplsLsrObjects)
- Traps (mplsLsrNotifications and mplsLsrNotifyPrefix)
- Conformance (mplsLsrConformance)

This Cisco implementation does not support the notifications defined in the MIB, nor does it support the labelStackTable or the trafficParamTable.

- [MPLS-LSR-MIB Tables, page 17](#)
- [Information from Scalar Objects, page 21](#)

MPLS-LSR-MIB Tables

The Cisco implementation of the MPLS-LSR-MIB supports four main tables:

- Interface configuration
- In-segment
- Out-segment
- Cross-connect

The MIB contains three supplementary tables to supply performance information. This implementation does not support the label stack and traffic parameter tables.

The following sections list the MPLS-LSR-MIB tables (main and supplementary), their functions, table objects that are supported, and table objects that are *not* supported.

MPLS interface configuration table (mplsInterfaceConfTable)

Provides information for each MPLS-capable interface on an LSR.

Supports:

- A unique interface index or zero
- Minimum and maximum values for an MPLS label received on the interface
- Minimum and maximum values for an MPLS label sent from the interface
- A value for an MPLS label sent from the interface
- Per platform (0) or per interface (1) setting
- The storage type

Does not support:

- The total usable bandwidth on the interface
- The difference between the total usable bandwidth and the bandwidth in use

MPLS interface performance table (mplsInterfacePerfTable)

Augments the MPLS interface configuration table.

Supports:

- The number of labels in the incoming direction in use
- The number of top-most labels in outgoing label stacks in use

Does not support:

- The number of top-most labels in outgoing label stacks in use
- The number of labeled packets discarded because no cross-connect entries exist
- The number of outgoing MPLS packets requiring fragmentation for transmission

MPLS in-segment table (mplsInSegmentTable)

Contains a description of incoming segments (labels) at an LSR and their associated parameters.

Administrative and operational status objects for this table control packet transmission. If administrative and operational status objects are down, the LSR does not forward packets. If these status objects are up, the LSR forwards packets.

Supports:

- A unique index identifier
- The incoming label
- The number of labels to pop from the incoming segment
- An address family number from the Internet Assigned Number Authority (IANA)
- A segment cross-connect entry association
- The segment owner
- The storage type
- The administrative status

- The operational status

**Note**

The administrative status and operational status are always up for inSegments in the Cisco implementation. Otherwise, these entries do not appear in the table.

Does not support:

- A pointer to a traffic parameter table entry (set to the default 0.0)

MPLS in-segment performance table (mplsInSegmentPerfTable)

Augments the MPLS in-segment table, providing performance information and counters for incoming segments on an LSR.

Supports:

- The number of 32-bit octets received
- The number of 64-bit octets received
- The time of the last system failure that corresponded to one or more incoming segment discontinuities

**Note**

The lastFailure parameter is set to zero because it has no meaning in the Cisco implementation.

Does not support:

- The total number of packets received
- The number of packets with errors
- The number of labeled packets discarded with no errors

MPLS out-segment table (mplsOutSegmentTable)

Contains a description of outgoing segments from an LSR and their associated parameters.

Administrative and operational status objects for this table control packet transmission. If administrative and operational status objects are down, the LSR does not forward packets. If these values are up, the LSR forwards packets.

Supports:

- A unique index identifier
- An interface index of the outgoing interface
- An indication of whether or not a top label is pushed onto the outgoing packet's label stack
- The label to push onto the outgoing packet's label stack (if the previous value is true)
- The next hop address type
- The IPv4 address of the next hop
- The segment cross-connect entry association
- The segment owner
- The storage type
- The administrative status
- The operational status

**Note**

The administrative and operational status entries are always up in the Cisco implementation. Otherwise, the administrative and operational status entries do not appear in the table.

Does not support:

- An IPv6 address of the next hop
- A pointer to a traffic parameter table entry (set to the default 0.0)

MPLS out-segment performance table (mplsOutSegmentPerfTable)

Augments the MPLS out-segment table, providing performance information and counters for outgoing segments on an LSR.

Supports:

- The number of 32-bit octets sent
- The number of 64-bit octets sent
- The time of the last system failure that corresponded to one or more outgoing segment discontinuities

Does not support:

- The number of packets sent
- The number of packets that could not be sent because of errors
- The number of packets discarded with no errors

MPLS cross-connect table (mplsXCTable)

Associates inSegments (labels) with outSegments (labels) to show the manager how the LSR is currently swapping these labels.

A row in this table consists of one cross-connect entry that is indexed by the cross-connect index, the interface index of the incoming segment, the incoming label, and the out-segment index.

The administrative and operational objects for this table control packet forwarding to and from a cross-connect entry (XCEntry). The administrative status and operational status are always up in the Cisco implementation. Otherwise, the LSR would not forward packets.

Supports:

- A unique index identifier for a group of cross-connect segments
- A label switched path (LSP) to which the cross-connect entry belongs
- An index to the MPLS label stack table that identifies the stack of labels to be pushed under the top label
- An indication whether or not to restore the cross-connect entry after a failure (the default value is false)
- The cross-connect owner
- The storage type
- The administrative status (if up)
- The operational status (if up)

**Note**

The administrative status and operational status are always up in the Cisco implementation. Otherwise, these status entries do not appear in the table.

Does not support:

- Tunnel IDs as label switched path (LSP) ID objects

Information from Scalar Objects

The MPLS-LSR-MIB supports several scalar objects. In the Cisco implementation of the MIB, the following scalar objects are hard-coded to the value indicated and are read-only objects:

- `mplsOutSegmentIndexNext (0)`--The value for the out-segment index when an LSR creates a new entry in the MPLS out-segment table. The 0 indicates that this is not implemented because modifications to this table are not allowed.
- `mplsXCIndexNext (0)`--The value for the cross-connect index when an LSR creates an entry in the MPLS cross-connect table. The 0 indicates that no unassigned values are available.
- `mplsMaxLabelDepth(2)`--The value for the maximum stack depth.
- `mplsLabelStackIndexNext (0)`--The value for the label stack index when an LSR creates entries in the MPLS label stack table. The 0 indicates that no unassigned values are available.
- `mplsTrafficParamIndexNext (0)`--The value for the traffic parameter index when an LSR creates entries in the MPLS traffic parameter table. The 0 indicates that no unassigned values are available.

The following scalar objects do not contain information for the MPLS-LSR-MIB and are coded as false:

- `mplsInSegmentTrapEnable (false)`--In-segment traps are not sent when this value is false.
- `mplsOutSegmentTrapEnable (false)`--Out-segment traps are not sent when this value is false.
- `mplsXCTrapEnable (false)`--Cross-connect traps are not sent when this value is false.

No trap information exists to support the MIB. Therefore, the following traps are not supported:

- `mplsInSegmentUp`
- `mplsInSegmentDown`
- `mplsOutSegmentUp`
- `mplsOutSegmentDown`
- `mplsXCUp`
- `mplsXCDown`

Linking Table Elements

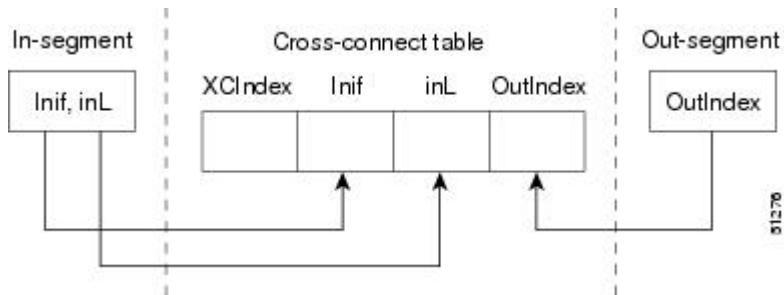
In the cross-connect table, cross-connect entries associate incoming segments and interfaces with outgoing segments and interfaces. The following objects index the cross-connect entry:

- `Cross-connect index`--A unique identifier for a group of cross-connect entries in the cross-connect table. In the Cisco implementation, this value is always the same as that for the `outSegmentIndex`, unless there is no label or if the label has been popped.
- `Interface index of the in-segment`--A unique index for an entry in the in-segment table that represents an incoming MPLS interface. The value 0 means platform wide, for any entries that apply to all interfaces.
- `Incoming label`--An entry in the in-segment table that represents the label on the incoming packet.

- Out-segment index--A unique identifier for an entry in the out-segment table that contains a top label for the outgoing packet's label stack and an interface index of the outgoing interface.

The figure below shows the links between the in-segment and the out-segment in the cross-connect table.

Figure 3 Cross-Connect Table Links



The table below shows the cross-connect table links you might see in the output from SNMP **get** operations on the MPLS-LSR-MIB objects that index a cross-connect entry. These objects include

- In-Segment Values--mplsInSegmentIfIndex and mplsInSegmentLabel
- Cross-Connect Entry--mplsXCIndex
- Out-Segment Values--mplsOutSegmentIndex

Table 6 MPLS LSR Output Showing Cross-Connect Table Links

In-Segment Values	Cross-Connect Entry	Out-Segment Values
0 ¹ , 1000	500 ² , 0, 1000, 0 Linking Table Elements, page 21	--
	501, 0, 1000, 501	501 = Pop (topLabel), Eth 1/5
	502, 0, 1000, 502	502 = Pop (topLabel), Eth, 1/1



Note

The OutSegmentIndex object is not the label. The label can be retrieved from the mplsOutSegmentTopLabel object.

Interface Configuration Table and Interface MIB Links

The MPLS interface configuration table lists interfaces that support MPLS technology. An LSR creates an entry dynamically in this table for each MPLS-capable interface. An interface becomes MPLS-capable when MPLS is enabled on that interface. A non-zero index for an entry in this table points to the ifIndex for the corresponding interface entry in the MPLS-layer in the ifTable of the Interfaces Group MIB.

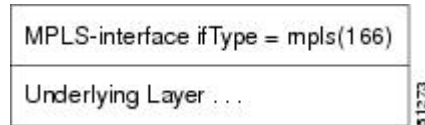
¹ All MPLS-enabled interfaces can receive incoming labels.

² For this implementation of the MPLS-LSR-MIB, the cross-connect index and the out-segment index are the same. If there is no outsegment, the value will be zero.

The ifTable contains information on each interface in the network. Its definition of an interface includes any sublayers of the internetwork layer of the interface. MPLS interfaces fit into this definition of an interface. Therefore, each MPLS-enabled interface is represented by an entry in the ifTable.

The interrelation of entries in the ifTable is defined by the interfaces stack group of the Interfaces Group MIB. The figure below shows how the stack table might appear for MPLS interfaces. The underlying layer refers to any interface that is defined for MPLS internetworking, for example, ATM, Frame Relay, or Ethernet.

Figure 4 Interface Group MIB Stack Table for MPLS Interfaces

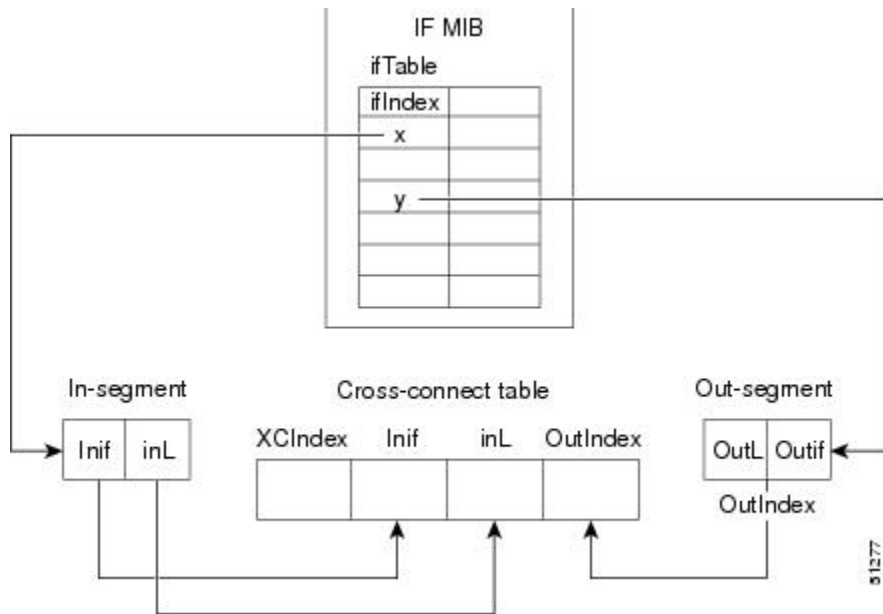


Note

Tunnel interfaces are included in the MPLS list for the current implementation.

The incoming and outgoing packets include a reference to the interface index for the ifTable of the Interfaces Group MIB. The figure below shows the links between MPLS-LSR-MIB objects and the Interfaces Group MIB.

Figure 5 MPLS-LSR-MIB and Interfaces Group MIB Links



- For the Interfaces Group MIB (IF MIB):
 - ifTable represents the MPLS interface table.
 - ifIndex represents the index to an entry in the MPLS interface table.
- For the In-segment:
 - Inif represents the interface on the incoming segment (references an index entry in the ifTable).

- inL represents the label on the incoming segment.
- For the Out-segment:
 - OutL represents the label on the outgoing segment.
 - Outif represents the interface on the outgoing segment (references an index entry in the ifTable).
- For the Cross-connect table:
 - XCIndex represents the index to an entry in the MPLS cross-connect table.
 - Inif represents the interface on the incoming segment.
 - inL represents the MPLS label on the incoming segment.
 - OutIndex represents an index to an entry in the MPLS out-segment table.

Using the MPLS-LSR-MIB

The MPLS-LSR-MIB enables you to display the contents of the MPLS Label Forwarding Information Base (LFIB). It gives you the same information that you can obtain using the CLI command **show mpls forwarding-table**.

However, the MPLS-LSR-MIB approach offers these advantages over the CLI command approach:

- A more efficient use of network bandwidth
- Greater interoperability among vendors
- Greater security (SMNP Version 3)

The following paragraphs describe the MPLS-LSR-MIB structure and show, through the use of an example, how the two approaches to the information display compare.

- [MPLS-LSR-MIB Structure, page 24](#)
- [CLI Commands and the MPLS-LSR-MIB, page 25](#)

MPLS-LSR-MIB Structure

MIB structure is represented by a tree hierarchy. Branches along the tree have short text strings and integers to identify them. Text strings describe object names, and integers allow computer software to encode compact representations of the names.

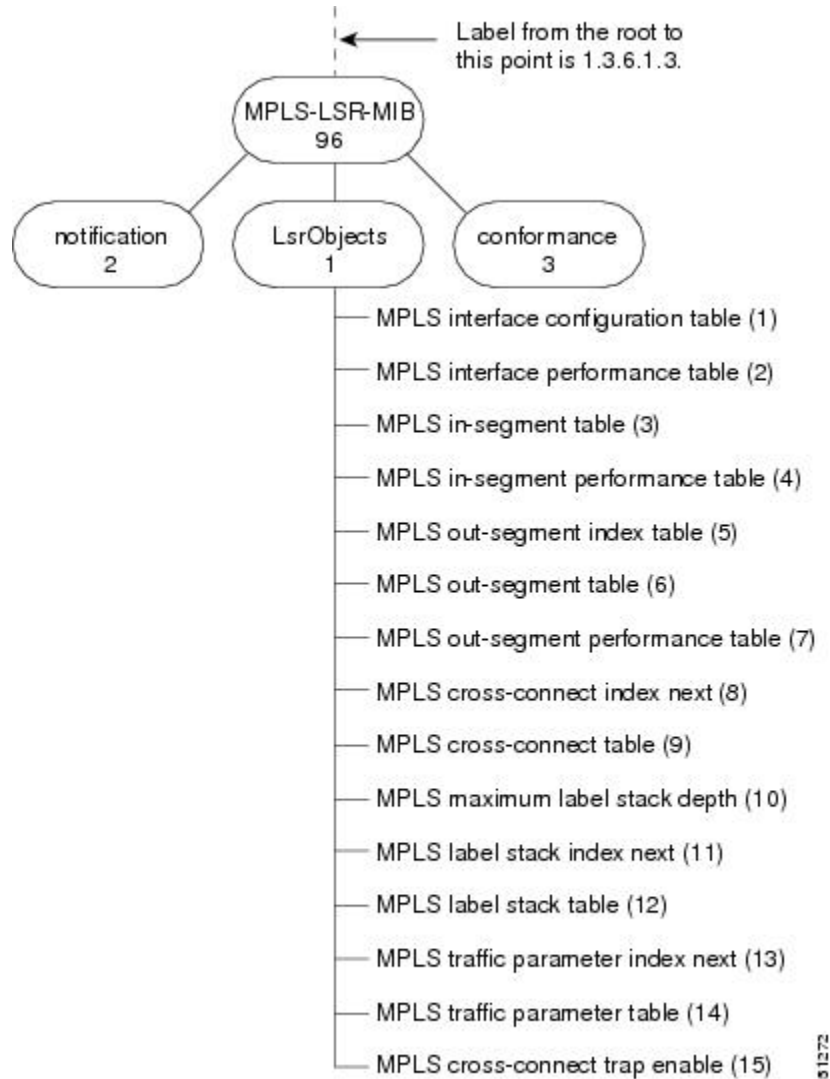
The MPLS-LSR-MIB falls on the experimental branch of the Internet MIB hierarchy. The experimental branch of the Internet MIB hierarchy is represented by the object identifier 1.3.6.1.3. This branch can also be represented by its object name *iso.org.dod.internet.experimental*. The MPLS-LSR-MIB is identified by the object name *mplsLsrMIB*, which is denoted by the number 96. Therefore, objects in the MPLS-LSR-MIB can be identified in either of the following ways:

- The object identifier--1.3.6.1.3.96.[MIB-variable]
- The object name--*iso.org.dod.internet.experimental.mplsLsrMIB.[MIB-variable]*

To display a *MIB-variable*, you enter an SNMP **get** command with an object identifier. Object identifiers are defined by the MPLS-LSR-MIB.

The figure below shows the position of the MPLS-LSR-MIB in the Internet MIB hierarchy.

Figure 6 *MPLS-LSR-MIB in the Internet MIB Hierarchy*



CLI Commands and the MPLS-LSR-MIB

The MPLS LFIB is the component of the Cisco MPLS subsystem that contains management information for LSRs. You can access this management information by means of either of the following:

- Using the **show mpls forwarding-table** CLI command
- Entering SNMP **get** commands on a network manager

The following examples show how you can gather LSR management information using both methods.

- [CLI Command Output, page 26](#)
- [MPLS-LSR-MIB Output, page 26](#)

CLI Command Output

A **show mpls forwarding-table** CLI command allows you to look at label forwarding information for a packet on a specific MPLS LSR.

```
Router# show mpls forwarding-table
Local  Outgoing  Prefix          Bytes Tag   Outgoing  Next Hop
Tag    Tag or VC  or Tunnel Id    Switched   interface
19     Pop Tag    10.3.4.0/24     0          Et1/4     10.22.23.23
22     23        14.14.14.14/32  0          AT2/0.1   point2point
      1/36      14.14.14.14/32  0          AT2/0.2   point2point
```

MPLS-LSR-MIB Output

SNMP commands on MIB objects also allow you to look at the label forwarding information for a specific MPLS LSR.

You can do a walk-through of the MIB by running a command such as **getmany -v2c public mplsLsrMIB** on a network manager where **getmany** does repeated SNMP **getnext** operations to retrieve the contents of the MPLS-LSR-MIB.

```
mplsXCOperStatus.9729.0.19.9729 = up(1)
mplsXCOperStatus.11265.0.22.11265 = up(1)
mplsXCOperStatus.11266.0.22.11266 = up(1)
```

You can continue to scan the output of the **getmany** command for the following (from the MPLS out-segment table):

- Out-segment's top label objects (mplsOutSegmentTopLabel)

```
mplsOutSegmentTopLabel.9729 = 3
mplsOutSegmentTopLabel.11265 = 23
mplsOutSegmentTopLabel.11266 = 65572
```



Note

65572 is 1/36 in label form (1 is the high-order 16 bits. 36 is the low-order 16 bits.)

- Out-segment's interface index (mplsOutSegmentIfIndex)

```
mplsOutSegmentIfIndex.9729 = 7
mplsOutSegmentIfIndex.11265 = 28
mplsOutSegmentIfIndex.11266 = 31
```

Benefits

The benefits described in the following paragraphs are available to you with the MPLS-LSR-MIB.

Troubleshooting LSR Problems

By monitoring the cross-connect entries and the associated incoming and outgoing segments, you can see which labels are installed and how they are being swapped. Use the MPLS-LSR-MIB in place of the **show mpls forwarding** CLI command.

Monitoring of LSR Traffic Loads

By monitoring interface and packet operations on an MPLS LSR, you can identify high- and low-traffic patterns, as well as traffic distributions.

Improvement of Network Performance

By identifying potentially high-traffic areas, you can set up load sharing to improve network performance.

Verification of LSR Configuration

By comparing results from SNMP **get** commands and the **show mpls forwarding** CLI command, you can verify your LSR configuration.

Displaying of Active Label Switched Paths

By monitoring the cross-connect entries and the associated incoming segments and outgoing segments, you can determine the active LSPs.

How to Configure the MPLS LSR MIB

- [Prerequisites, page 27](#)
- [Enabling the SNMP Agent, page 27](#)
- [Verifying That the SNMP Agent Has Been Enabled, page 28](#)

Prerequisites

The MPLS-LSR-MIB requires the following:

- SNMP installed and enabled on the LSR
- MPLS enabled on the LSR
- 60K of memory

**Note**

Additional capacity is not required for runtime dynamic random-access memory (DRAM).

Enabling the SNMP Agent

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

SUMMARY STEPS

1. **enable**
2. **show running-config**
3. **configure terminal**
4. **snmp-server community** *string* [**view** *view-name*] [**ro**] [*number*]
5. **end**
6. **copy running-config startup-config**

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 <code>enable</code></p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
<p>Step 2 <code>show running-config</code></p> <p>Example:</p> <pre>Router# show running-config</pre>	<p>Displays the running configuration of the router to determine if an SNMP agent is already running on the device.</p> <p>If no SNMP information is displayed, continue with the next step.</p> <p>If any SNMP information is displayed, you can modify the information or change it as desired.</p>
<p>Step 3 <code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 4 <code>snmp-server community string [view view-name] [ro] [number]</code></p> <p>Example:</p> <pre>Router(config)# snmp-server community public ro</pre>	<p>Configures read-only (ro) SNMP community strings.</p> <p>This command enables the SNMP agent and permits any SNMP manager to access all objects with read-only permission using the community string public.</p>
<p>Step 5 <code>end</code></p> <p>Example:</p> <pre>Router(config)# end</pre>	<p>Exits to privileged EXEC mode.</p>
<p>Step 6 <code>copy running-config startup-config</code></p> <p>Example:</p> <pre>Router# copy running-config startup-config</pre>	<p>Copies the modified SNMP configuration into router NVRAM, permanently saving the SNMP settings.</p> <p>When you are working with Cisco IOS Release 10.3 or earlier, use the write memory command.</p>

Verifying That the SNMP Agent Has Been Enabled

To verify that the SNMP agent has been enabled, perform the following steps:

SUMMARY STEPS

1. Access the router through a Telnet session:
2. Enter privileged mode:
3. Display the running configuration and look for SNMP information:

DETAILED STEPS

Step 1 Access the router through a Telnet session:

Example:

```
Prompt# telnet xxx.xxx.xxx.xxx
```

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

Step 2 Enter privileged mode:

Example:

```
Router# enable
```

Step 3 Display the running configuration and look for SNMP information:

Example:

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

If you see any “snmp-server” statements, SNMP has been enabled on the router.

Configuration Examples for the MPLS LSR MIB

The following example shows how to enable an SNMP agent.

```
configure terminal
snmp-server community
```

In the following example, SNMPv1 and SNMPv2C are enabled. The configuration permits any SNMP manager to access all objects with read-only permissions using the community string *public*.

```
configure terminal
snmp-server community public
```

In the following example, read-only access is allowed for all objects to members of access list 4 that specify the *comaccess* community string. No other SNMP managers have access to any objects.

```
configure terminal
nmp-server community comaccess ro 4
```

Additional References

Related Documents

Related Topic	Document Title
Configuring SNMP using Cisco IOS software	<ul style="list-style-type: none"> <i>Network Management Configuration Guide . Configuring SNMP Support</i> <i>Network Management Command Reference, SNMP Commands</i>

Standards

Standard	Title
draft-ietf-mpls-lsr-mib-05.txt	MPLS Label Switch Router Management Information Base Using SMIV2
draft-ietf-mpls-arch-07.txt	Multiprotocol Label Switching Architecture

MIBs

MIBs	MIBs Link
<ul style="list-style-type: none"> MPLS Label Switching Router MIB (MPLS-LSR-MIB) 	<p>To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL:</p> <p>http://www.cisco.com/go/mibs</p>

RFCs

RFCs	Title
The LSR implementation supporting the MPLS-LSR-MIB is in full compliance with all provisions of Section 10 of RFC 2026.	<i>The Internet Standards Process</i>

Technical Assistance

Description	Link
<p>The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.</p>	http://www.cisco.com/techsupport
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p>	
<p>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.</p>	
<p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	

Command Reference

This feature uses no new or modified commands.

Glossary

cross-connect (XC) --An association of in-segments and incoming Multiprotocol Label Switching (MPLS) interfaces to out-segments and outgoing MPLS interfaces.

IETF --Internet Engineering Task Force. A task force (consisting of more than 80 working groups) that is developing standards for the Internet and the IP suite of protocols.

inSegment --A label on an incoming packet that is used to determine the forwarding of the packet.

Internet Engineering Task Force --See IETF.

label --A short, fixed length identifier that is used to determine the forwarding of a packet.

Label Distribution Protocol --See LDP.

label switched path --See LSP.

label switching --Describes the forwarding of IP (or other network layer) packets by a label swapping algorithm based on network layer routing algorithms. The forwarding of these packets uses the exact match algorithm and rewrites the label.

label switch router --See LSR.

LDP --Label Distribution Protocol. A standard protocol that operates between Multiprotocol Label Switching (MPLS)-enabled routers to negotiate the labels (addresses) used to forward packets. The Cisco proprietary version of this protocol is the Tag Distribution Protocol (TDP).

LSP --label switched path. A sequence of hops in which a packet travels from one router to another router by means of label switching mechanisms. A label switched path can be established dynamically, based on normal routing mechanisms, or through configuration.

LSR --label switch router. A device that forwards Multiprotocol Label Switching (MPLS) packets based on the value of a fixed-length label encapsulated in each packet.

Management Information Base --See MIB.

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

MPLS --Multiprotocol Label Switching. A switching method that forwards IP traffic through use of a label. This label instructs the routers and the switches in the network where to forward the packets. The forwarding of MPLS packets is based on preestablished IP routing information.

MPLS interface --An interface on which Multiprotocol Label Switching (MPLS) traffic is enabled.

Multiprotocol Label Switching --See MPLS.

notification request --A message sent by a Simple Network Management Protocol (SNMP) agent to a network management station, console, or terminal, indicating that a significant event occurred. SNMP notification requests are more reliable than traps, because a notification request from an SNMP agent requires that the SNMP manager acknowledge receipt of the notification request. The manager replies with an SNMP response protocol data unit (PDU). If the manager does not receive a notification message from an SNMP agent, it does not send a response. If the sender (SNMP agent) never receives a response, the notification request can be sent again.

outSegment --A label on an outgoing packet.

Simple Network Management Protocol --See SNMP.

SNMP --Simple Network Management Protocol. A management protocol used almost exclusively in TCP/IP networks. SNMP provides a means for monitoring and controlling network devices, and for managing configurations, statistics collection, performance, and security.

trap --A message sent by a Simple Network Management Protocol (SNMP) agent to a network management station, console, or terminal, indicating that a significant event occurred. Traps are less reliable than notification requests, because the receiver does not send an acknowledgment when it receives a trap. The sender cannot determine if the trap was received.



Note

Refer to the Cisco *Dictionary of Internetworking Terms and Acronyms* for terms not included in this glossary.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks.

Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

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



MPLS Label Distribution Protocol MIB Version 8 Upgrade

The MPLS Label Distribution Protocol (LDP) MIB Version 8 Upgrade feature enhances the LDP MIB to support the Internet Engineering Task Force (IETF) draft Version 8.

- [Finding Feature Information, page 35](#)
- [Prerequisites for MPLS LDP MIB Version 8 Upgrade, page 35](#)
- [Restrictions for MPLS LDP MIB Version 8 Upgrade, page 35](#)
- [Information About MPLS LDP MIB Version 8 Upgrade, page 36](#)
- [How to Configure MPLS LDP MIB Version 8 Upgrade, page 58](#)
- [Configuration Examples for MPLS LDP MIB Version 8 Upgrade, page 71](#)
- [Additional References, page 72](#)
- [Feature Information for MPLS LDP MIB Version 8 Upgrade, page 73](#)
- [Glossary, page 75](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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.

Prerequisites for MPLS LDP MIB Version 8 Upgrade

- Simple Network Management Protocol (SNMP) must be installed and enabled on the label switch routers (LSRs).
- Multiprotocol Label Switching (MPLS) must be enabled on the LSRs.
- LDP must be enabled on the LSRs.

Restrictions for MPLS LDP MIB Version 8 Upgrade

This implementation of the MPLS LDP MIB is limited to read-only (RO) permission for MIB objects, except for MIB object *mplsLdpSessionUpDownTrapEnable*, which 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.

For a description of notification events, see the Events Generating MPLS LDP MIB Notifications in MPLS LDP MIB Version 8 Upgrade 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.

The following tables are not implemented in this feature:

- *mplsLdpEntityFrParmsTable*
- *mplsLdpEntityConfFrLRTable*
- *mplsLdpFrameRelaySesTable*
- *mplsFecTable*
- *mplsLdpSesInLabelMapTable*
- *mplsXCsfecsTable*
- *mplsLdpSesPeerAddrTable*

Information About MPLS LDP MIB Version 8 Upgrade

- [Feature Design of MPLS LDP MIB Version 8 Upgrade, page 36](#)
- [Enhancements in Version 8 of the MPLS LDP MIB, page 38](#)
- [Benefits of MPLS LDP MIB Version 8 Upgrade, page 38](#)
- [Description of MPLS LDP MIB Elements for MPLS LDP MIB Version 8 Upgrade, page 38](#)
- [Events Generating MPLS LDP MIB Notifications in MPLS LDP MIB Version 8 Upgrade, page 42](#)
- [MIB Tables in MPLS LDP MIB Version 8 Upgrade, page 43](#)

Feature Design of MPLS LDP MIB Version 8 Upgrade

MPLS is a packet forwarding technology that uses a short, fixed-length value called a label in packets to specify the next hop for packet transport through an MPLS network by means of label switch 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 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 the two negotiate basic operating procedures. 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. LDP then creates an active LDP session between the two LSRs to effect the exchange of label binding information. When this process is carried to completion with respect to all of the LSRs in an MPLS network, the result 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 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 code has 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, and you can use those objects to accomplish a variety of network management tasks. All the objects in the MPLS LDP MIB follow the conventions defined in the 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 evolving and is soon expected to be a standard. Accordingly, the MPLS LDP MIB will be implemented in such a way that it tracks the evolution of this IETF document.

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

The extensive Cisco label switching capabilities provide an integrated approach to managing the large volumes of traffic carried by 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 resistance of the network to link or node failures.

The MPLS Label Distribution Protocol MIB Version 8 Upgrade supports the following functions:

- Tag Distribution Protocol (TDP) (This protocol might not be supported in all software releases.)
- Generation and sending of event notification messages that 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 event notification messages are to be sent to serve network administrative and management purposes
- Storage of the configuration pertaining to an event notification message in NVRAM of the NMS

The structure of the MPLS LDP MIB conforms to Abstract Syntax Notation One (ASN.1), so the MIB forms 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 and GETNEXT operations.

**Note**

Because the MPLS LDP MIB was not given an Internet Assigned Numbers Authority (IANA) experimental object identifier (OID) at the time of its implementation, Cisco chose to implement the MIB under the `ciscoExperimental` OID number, as follows: `ciscoExperimental 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 replace all objects in the MIB under the `ciscoExperimental` OID and reposition the objects under the IANA Experimental OID.

Enhancements in Version 8 of the MPLS LDP MIB

Version 8 of the MPLS LDP MIB contains the following enhancements:

- TDP support (This protocol might not be supported in all software releases.)
- Upgraded objects
- New indexing that is no longer based on the number of sessions
- Multiple SNMP context support for Virtual Private Networks (VPNs)

Benefits of MPLS LDP MIB Version 8 Upgrade

- Supports TDP and LDP (TDP might not be supported in all software releases.)
- Establishes LDP sessions between peer devices in an MPLS network
- Retrieves 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 virtual path identifier/virtual channel identifier (VPI/VCI) pairs to be used in forming labels
- Gathers statistics related to LDP operations, such as error counters.
- Monitors the time remaining for hello adjacencies
- Monitors the characteristics and status of LDP peers, such as:
 - Internetwork layer address of LDP peers
 - Loop detection of the LDP peers
 - Default MTU of the LDP peer
 - Number of seconds the LDP peer proposes as the value of the keepalive interval
- Monitors the characteristics and status of LDP sessions, such as:
 - Displaying the error counters.
 - 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)
 - Displaying the label ranges for platform-wide and interface-specific sessions
 - Displaying the ATM parameters.

Description of MPLS LDP MIB Elements for MPLS LDP MIB Version 8 Upgrade

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; describes a potential session.
- 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 that affirms the state of two LSRs in an MPLS network as being adjacent to each other (that is, as being LDP peers). When the neighbor is discovered, the neighbor becomes a hello adjacency. An LDP session can be established with the hello adjacency. After the session is established, label bindings can be exchanged between the LSRs.

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 in the database. This database reflects the current state of MPLS LDP operations in the network. You can access this network management information database 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, page 39](#)
- [LDP Sessions and Peers, page 40](#)
- [LDP Hello Adjacencies, page 41](#)

LDP Entities

An LDP entity is uniquely identified by an LDP identifier that consists of the `mplsLdpEntityLdpId` and the `mplsLdpEntityIndex` (see the figure below).

- The `mplsLdpEntityLdpId` consists of the local LSR ID (four octets) and the label space ID (two octets). The label space ID identifies a specific label space available within the LSR.
- The `mplsLdpEntityIndex` consists of the IP address of the peer active hello adjacency, which is the 32-bit representation of the IP address assigned to the peer LSR.

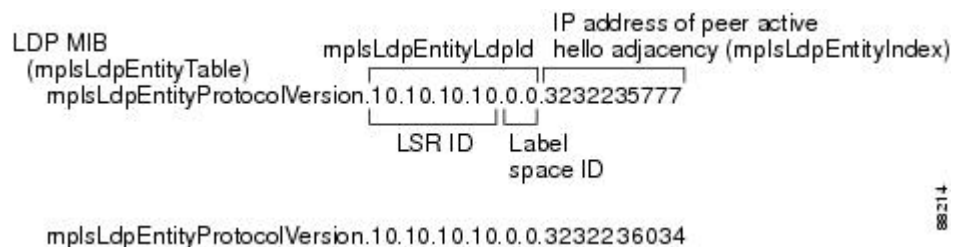
The `mplsLdpEntityProtocolVersion` is a sample object from the `mplsLdpEntityTable`.

The figure shows the following indexing:

- `mplsLdpEntityLdpId` = 10.10.10.10.0.0
- LSR ID = 10.10.10.10
- Label space ID = 0.0

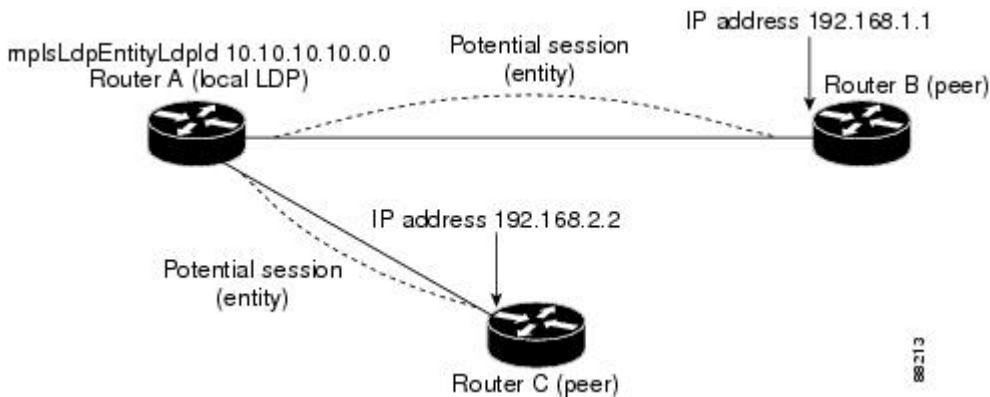
The `mplsLdpEntityLdpId` or the LDP ID consists of the LSR ID and the label space ID.

- The IP address of peer active hello adjacency or the `mplsLdpEntityIndex` = 3232235777, which is the 32-bit representation of the IP address assigned to the peer's active hello adjacency.



An LDP entity represents a label space that has the potential for a session with an LDP peer. An LDP entity is set up when a hello adjacency receives a hello message from an LDP peer.

In the figure below, Router A has potential sessions with two remote peers, Routers B and C. The `mplsLdpEntityLdpId` is 10.10.10.10.0.0, and the IP address of the peer active hello adjacency (`mplsLdpEntityIndex`) is 3232235777, which is the 32-bit representation of the IP address 192.168.1.1 for Router B.



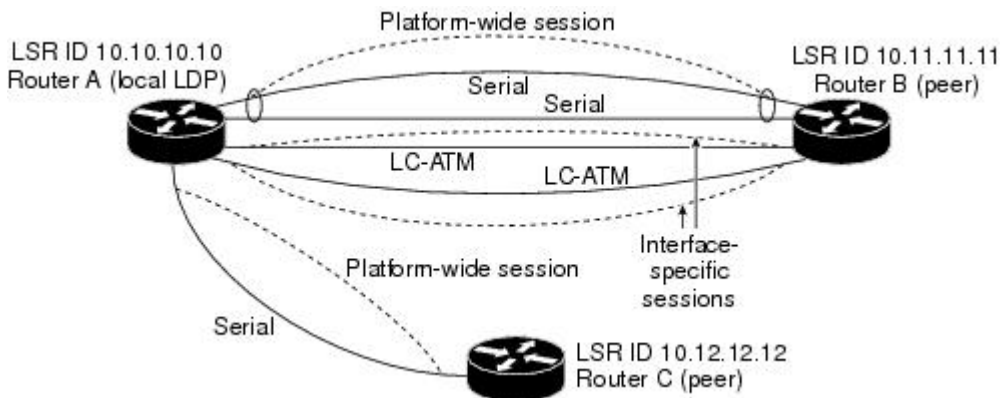
LDP Sessions and Peers

LDP sessions exist between local entities and remote peers for the purpose 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 an LDP instance that communicates across one or more network links with a single LDP peer. LDP supports the following types of sessions:

- **Interface-specific**--An interface-specific session uses interface resources for label space distributions. For example, each label-controlled ATM (LC-ATM) interface uses its own VPIs/VCIs for label space distributions. Depending on its configuration, an LDP platform can support zero, one, or more interface-specific sessions. Each LC-ATM interface has its own interface-specific label space and a nonzero label space ID.
- **Platform-wide**--An LDP platform supports a single platform-wide session for use by all interfaces that can share the same global label space. For Cisco platforms, all interface types except LC-ATM use the platform-wide session and have a label space ID of zero.

When a session is established between two peers, entries are created in the `mplsLdpPeerTable` and the `mplsLdpSessionTable` because they have the same indexing.

In the figure below, Router A has two remote peers, Routers B and C. Router A has a single platform-wide session that consists of two serial interfaces with Router B and another platform-wide session with Router C. Router A also has two interface-specific sessions with Router B.



The figure below shows entries that correspond to the `mplsLdpPeerTable` and the `mplsLdpSessionTable` in the figure above.

In the figure below, `mplsLdpSesState` is a sample object from the `mplsLdpSessionTable` on Router A. There are four `mplsLdpSesState` sample objects shown (top to bottom). The first object represents a platform-wide session associated with two serial interfaces. The next two objects represent interface-specific sessions for the LC-ATM interfaces on Routers A and B. These interface-specific sessions have nonzero peer label space IDs. The last object represents a platform-wide session for the next peer, Router C.

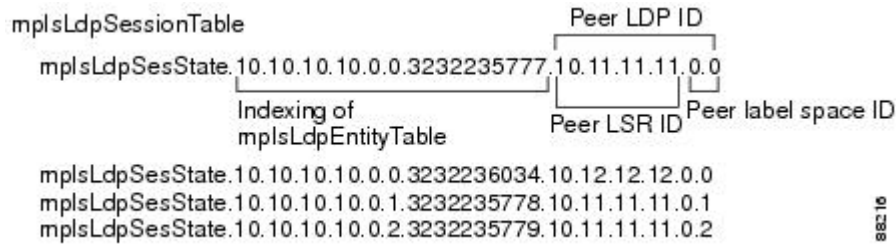
The indexing is based on the entries in the `mplsLdpEntityTable`. It begins with the indexes of the `mplsLdpEntityTable` and adds the following:

- Peer LDP ID = 10.11.11.11.0.0

The peer LDP ID consists of the peer LSR ID (four octets) and the peer label space ID (two octets).

- Peer LSR ID = 10.11.11.11
- Peer label space ID = 0.0

The peer label space ID identifies a specific peer label space available within the LSR.



LDP Hello Adjacencies

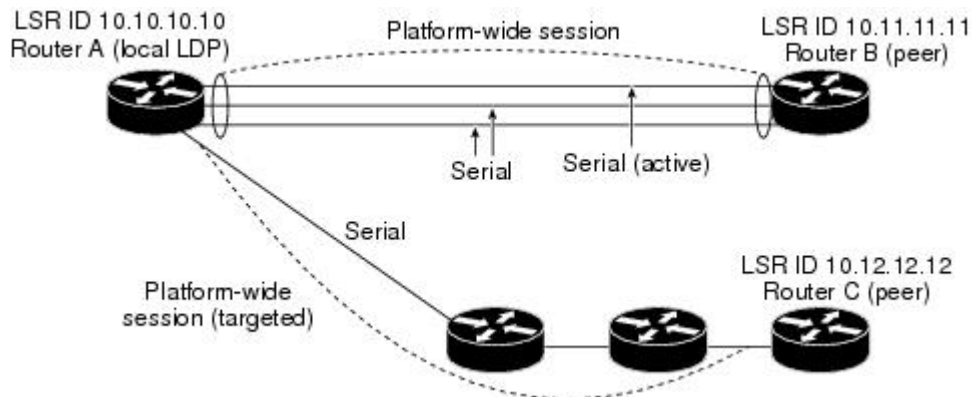
An LDP hello adjacency is a network link between a router and its peers. An LDP hello adjacency enables two adjacent peers to exchange label binding information.

An LDP hello adjacency exists for each link on which LDP runs. Multiple LDP hello adjacencies exist whenever there is more than one link in a session between a router and its peer, such as in a platform-wide session.

A hello adjacency is considered active if it is currently engaged in a session, or nonactive if it is not currently engaged in a session.

A targeted hello adjacency is not directly connected to its peer and has an unlimited number of hops between itself and its peer. A linked hello adjacency is directly connected between two routers.

In the figure below, Router A has two remote peers, Routers B and C. Router A has a platform-wide session with Router B that consists of three serial interfaces, one of which is active and another platform-wide (targeted) session with Router C.



The figure below shows entries in the `mplsLdpHelloAdjacencyTable`. There are four `mplsLdpHelloAdjHoldTimeRem` sample objects (top to bottom). They represent the two platform-wide sessions and the four serial links shown in the figure above.

The indexing is based on the `mplsLdpSessionTable`. When the `mplsLdpHelloAdjIndex` enumerates the different links within a single session, the active link is `mplsLdpHelloAdjIndex = 1`.

```

mplsLdpHelloAdjacencyTable
  mplsLdpHelloAdjHoldTimeRem.10.10.10.10.0.0.3232235777.10.11.11.11.0.0.1
                                     Indexing of mplsLdpSessionTable      mplsLdpHelloAdjIndex
  mplsLdpHelloAdjHoldTimeRem.10.10.10.10.0.0.3232235777.10.11.11.11.0.0.2
  mplsLdpHelloAdjHoldTimeRem.10.10.10.10.0.0.3232235777.10.11.11.11.0.0.3
  mplsLdpHelloAdjHoldTimeRem.10.10.10.10.0.0.3232236034.10.12.12.12.0.0.1

```

Events Generating MPLS LDP MIB Notifications in MPLS LDP MIB Version 8 Upgrade

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 the network.

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.

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 different path vector limits.



Note

This notification is generated only if the distribution method is downstream-on-demand.

- `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 and cannot be changed.

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. 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.

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 about a condition in the network that might 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 types of LDP feature support

MIB Tables in MPLS LDP MIB Version 8 Upgrade

Version 8 of the MPLS LDP MIB consists of the following tables:

- `mplsLdpEntityTable` --Contains entries for every active LDP hello adjacency. Nonactive hello adjacencies appear in the `mplsLdpHelloAdjacencyTable`, rather than this table. This table is indexed by the local LDP identifier for the interface and the IP address of the peer active hello adjacency.

The advantage of showing the active hello adjacency instead of sessions in this table is that the active hello adjacency can exist even if an LDP session is not active (cannot be established). Previous implementations of the IETF MPLS-LDP MIB used sessions as the entries in this table. This approach was inadequate because as sessions went down, the entries in the entity table would disappear completely because the agent code could no longer access them. This resulted in the MIB failing to provide information about failed LDP sessions.

Directed adjacencies are also shown in this table. These entries, however, are always up administratively (`adminStatus`) and operationally (`operStatus`), because the adjacencies disappear if the directed session fails. Nondirected adjacencies might disappear from the MIB on some occasions, because adjacencies are deleted if the underlying interface becomes operationally down, for example.

- `mplsLdpEntityConfGenLRTable` --Contains entries for every LDP-enabled interface that is in the global label space. (For Cisco, this applies to all interfaces except LC-ATM. LC-ATM entities are shown in the `mplsLdpEntityConfAtmLRTable` instead.) Indexing is the same as it is for the `mplsLdpEntityTable`, except two indexes have been added, `mplsLdpEntityConfGenLRMin` and `mplsLdpEntityConfGenLRMax`. These additional indexes allow more than one label range to be defined. However, in the current Cisco implementation, only one global label range is allowed.
- `mplsLdpEntityAtmParmsTable` --Contains entries for every LDP-enabled LC-ATM interface. This table is indexed the same as the `mplsLdpEntityTable` although only LC-ATM interfaces are shown.
- `mplsLdpEntityConfAtmLRTable` --Contains entries for every LDP-enabled LC-ATM interface. Indexing is the same as it is for the `mplsLdpEntityTable`, except two indexes have been added, `mplsLdpEntityConfAtmLRMinVpi` and `mplsLdpEntityConfAtmLRMinVci`. These additional indexes

allow more than one label range to be defined. However, in the current Cisco implementation, only one label range per LC-ATM interface is allowed.

- mplsLdpEntityStatsTable --Augments the mplsLdpEntityTable and shares the exact same indexing for performing GET and GETNEXT operations. This table shows additional statistics for entities.
 - mplsLdpPeerTable --Contains entries for all peer sessions. This table is indexed by the local LDP identifier of the session, the IP address of the peer active hello adjacency, and the peer's LDP identifier.
 - mplsLdpHelloAdjacencyTable --Contains entries for all hello adjacencies. This table is indexed by the local LDP identifier of the associated session, the IP address of the peer active hello adjacency, the LDP identifier for the peer, and an arbitrary index that is set to the list position of the adjacency.
 - mplsLdpSessionTable --Augments the mplsLdpPeerTable and shares the same indexing for performing GET and GETNEXT operations. This table shows all sessions.
 - mplsLdpAtmSesTable --Contains entries for LC-ATM sessions. Indexing is the same as it is for the mplsLdpPeerTable, except two indexes have been added, mplsLdpSesAtmLRLowerBoundVpi and mplsLdpSesAtmLRLowerBoundVci. These additional indexes allow more than one label range to be defined. However, in the current Cisco implementation, only one label range per LC-ATM interface is allowed.
 - mplsLdpSesStatsTable --Augments the mplsLdpPeerTable and shares the exact same indexing for performing GET and GETNEXT operations. This table shows additional statistics for sessions.
- [mplsLdpEntityTable](#), page 44
 - [mplsLdpEntityConfGenLRTable](#), page 47
 - [mplsLdpEntityAtmParmsTable](#), page 48
 - [mplsLdpEntityConfAtmLRTable](#), page 49
 - [mplsLdpEntityStatsTable](#), page 49
 - [mplsLdpPeerTable](#), page 51
 - [mplsLdpHelloAdjacencyTable](#), page 51
 - [mplsLdpSessionTable](#), page 52
 - [mplsLdpAtmSesTable](#), page 53
 - [mplsLdpSesStatsTable](#), page 53
 - [VPN Contexts in MPLS LDP MIB Version 8 Upgrade](#), page 54
 - [SNMP Context](#), page 54
 - [VPN Aware LDP MIB Sessions](#), page 54
 - [VPN Aware LDP MIB Notifications](#), page 56

mplsLdpEntityTable

The table below lists the mplsLdpEntityTable objects and their descriptions.

Table 7 *mplsLdpEntityTable Objects and Descriptions*

Object	Description
mplsLdpEntityEntry	Represents an LDP entity, which is a potential session between two peers.

Object	Description
mplsLdpEntityLdpId	The LDP identifier (not accessible) consists of the local LSR ID (four octets) and the label space ID (two octets).
mplsLdpEntityIndex	A secondary index that identifies this row uniquely. It consists of the IP address of the peer active hello adjacency, which is the 32-bit representation of the IP address assigned to the LSR (not accessible).
mplsLdpEntityProtocolVersion	The version number of the LDP protocol to be used in the session initialization message.
mplsLdpEntityAdminStatus	The administrative status of this LDP entity is always up. If the hello adjacency fails, this entity disappears from the mplsLdpEntityTable.
mplsLdpEntityOperStatus	The operational status of this LDP entity. Values are unknown(0), enabled(1), and disabled(2).
mplsLdpEntityTcpDscPort	The TCP discovery port for LDP or TDP. The default value is 646 (LDP).
mplsLdpEntityUdpDscPort	The UDP discovery port for LDP or TDP. The default value is 646 (LDP).
mplsLdpEntityMaxPduLength	The maximum PDU length that is sent in the common session parameters of an initialization message.
mplsLdpEntityKeepAliveHoldTimer	The two-octet value that is the proposed keepalive hold time for this LDP entity.
mplsLdpEntityHelloHoldTimer	The two-octet value that is the proposed hello hold time for this LDP entity.
mplsLdpEntityInitSesThreshold	The threshold for notification when this entity and its peer are engaged in an endless sequence of initialization messages. The default value is 8 and cannot be changed by SNMP or CLI.
mplsLdpEntityLabelDistMethod	The specified method of label distribution for any given LDP session. Values are downstreamOnDemand(1) and downstreamUnsolicited(2).
mplsLdpEntityLabelRetentionMode	Can be configured to use either conservative(1) for LC-ATM or liberal(2) for all other interfaces.

Object	Description
mplsLdpEntityPVLMisTrapEnable	<p>Indicates whether the mplsLdpPVLMismatch trap should be generated.</p> <p>If the value is enabled(1), the trap is generated. If the value is disabled(2), the trap is not generated. The default is disabled(2).</p> <p>Note The mplsLdpPVLMismatch trap is generated only if mplsLdpEntityLabelDistMethod is downstreamOnDemand(1).</p>
mplsLdpEntityPVL	<p>If the value of this object is 0, loop detection for path vectors is disabled. Otherwise, if this object has a value greater than zero, loop detection for path vectors is enabled, and the path vector limit is this value.</p> <p>Note The mplsLdpEntityPVL object is non-zero only if mplsLdpEntityLabelDistMethod is downstreamOnDemand(1).</p>
mplsLdpEntityHopCountLimit	<p>If the value of this object is 0, loop detection using hop counters is disabled.</p> <p>If the value of this object is greater than 0, loop detection using hop counters is enabled, and this object specifies this entity's maximum allowable value for the hop count.</p> <p>Note The mplsLdpEntityHopCountLimit object is non-zero only if mplsLdpEntityLabelDistMethod is downstreamOnDemand(1).</p>
mplsLdpEntityTargPeer	<p>If this LDP entity uses a targeted adjacency, this object is set to true(1). The default value is false(2).</p>
mplsLdpEntityTargPeerAddrType	<p>The type of the internetwork layer address used for the extended discovery. This object indicates how the value of mplsLdpEntityTargPeerAddr is to be interpreted.</p>
mplsLdpEntityTargPeerAddr	<p>The value of the internetwork layer address used for the targeted adjacency.</p>

Object	Description
mplsLdpEntityOptionalParameters	<p>Specifies the optional parameters for the LDP initialization message. If the value is generic(1), no optional parameters are sent in the LDP initialization message associated with this entity.</p> <p>LC-ATM uses atmParameters(2) to specify that a row in the mplsLdpEntityAtmParmsTable corresponds to this entry.</p> <p>Note Frame Relay parameters are not supported.</p>
mplsLdpEntityDiscontinuityTime	<p>The value of sysUpTime on the most recent occasion when one or more of this entity's counters suffered a discontinuity. The relevant counters are the specific instances of any Counter32 or Counter64 object contained in the mplsLdpEntityStatsTable that are associated with this entity. If no such discontinuities have occurred since the last reinitialization of the local management subsystem, this object contains a 0 value.</p>
mplsLdpEntityStorType	<p>The storage type for this entry is a read-only implementation that is always volatile.</p>
mplsLdpEntityRowStatus	<p>This object is a read-only implementation that is always active.</p>

mplsLdpEntityConfGenLRTable

The table below lists the mplsLdpEntityConfGenLRTable objects and their descriptions.

Table 8 *mplsLdpEntityConfGenLRTable Objects and Descriptions*

Object	Description
mplsLdpEntityConfGenLREntry	<p>A row in the LDP Entity Configurable Generic Label Range table. One entry in this table contains information on a single range of labels; the range is defined by an upper boundary (VPI/VCI pair) and a lower boundary (VPI/VCI pair).</p> <p>The current implementation supports one label range per entity.</p>
mplsLdpEntityConfGenLRMin	<p>The minimum label configured for this range (not accessible).</p>
mplsLdpEntityConfGenLRMax	<p>The maximum label configured for this range (not accessible).</p>

Object	Description
mplsLdpEntityConfGenIfIndxOrZero	This value represents the SNMP IF-MIB index for the platform-wide entity. If the active hello adjacency is targeted, the value is 0.
mplsLdpEntityConfGenLRStorType	The storage type for this entry is a read-only implementation that is always volatile.
mplsLdpEntityConfGenLRRowStatus	This object is a read-only implementation that is always active.

mplsLdpEntityAtmParamsTable

The table below lists the mplsLdpEntityAtmParamsTable objects and their descriptions.

Table 9 *mplsLdpEntityAtmParamsTable Objects and Descriptions*

Object	Description
mplsLdpEntityAtmParamsEntry	Represents the ATM parameters and ATM information for this LDP entity.
mplsLdpEntityAtmIfIndxOrZero	This value represents the SNMP IF-MIB index for the interface-specific LC-ATM entity.
mplsLdpEntityAtmMergeCap	Denotes the merge capability of this entity.
mplsLdpEntityAtmLRComponents	Number of label range components in the initialization message. This also represents the number of entries in the mplsLdpEntityConfAtmLRTable that correspond to this entry.
mplsLdpEntityAtmVcDirectionality	If the value of this object is bidirectional(0), a given VCI within a given VPI is used as a label for both directions independently of one another. If the value of this object is unidirectional(1), a given VCI within a VPI designates one direction.
mplsLdpEntityAtmLsrConnectivity	The peer LSR can be connected indirectly by means of an ATM VP, so that the VPI values can be different on the endpoints. For that reason, the label must be encoded entirely within the VCI field. Values are direct(1), the default, and indirect(2).
mplsLdpEntityDefaultControlVpi	The default VPI value for the non-MPLS connection.
mplsLdpEntityDefaultControlVci	The default VCI value for the non-MPLS connection.

Object	Description
mplsLdpEntityUnlabTrafVpi	VPI value of the VCC supporting unlabeled traffic. This non-MPLS connection is used to carry unlabeled (IP) packets.
mplsLdpEntityUnlabTrafVci	VCI value of the VCC supporting unlabeled traffic. This non-MPLS connection is used to carry unlabeled (IP) packets.
mplsLdpEntityAtmStorType	The storage type for this entry is a read-only implementation that is always volatile.
mplsLdpEntityAtmRowStatus	This object is a read-only implementation that is always active.

mplsLdpEntityConfAtmLRTable

The table below lists the mplsLdpEntityConfAtmLRTable objects and their descriptions.

Table 10 *mplsLdpEntityConfAtmLRTable Objects and Descriptions*

Object	Description
mplsLdpEntityConfAtmLREntry	A row in the LDP Entity Configurable ATM Label Range Table. One entry in this table contains information on a single range of labels; the range is defined by an upper boundary (VPI/VCI pair) and a lower boundary (VPI/VCI pair). This is the same data used in the initialization message. This label range should overlap the label range of the peer.
mplsLdpEntityConfAtmLRMinVpi	The minimum VPI number configured for this range (not accessible).
mplsLdpEntityConfAtmLRMinVci	The minimum VCI number configured for this range (not accessible).
mplsLdpEntityConfAtmLRMaxVpi	The maximum VPI number configured for this range (not accessible).
mplsLdpEntityConfAtmLRMaxVci	The maximum VCI number configured for this range (not accessible).
mplsLdpEntityConfAtmLRStorType	The storage type for this entry is a read-only implementation that is always volatile.
mplsLdpEntityConfAtmLRRowStatus	This object is a read-only implementation that is always active.

mplsLdpEntityStatsTable

The table below lists the mplsLdpEntityStatsTable objects and their descriptions.

Table 11 *mplsLdpEntityStatsTable Objects and Descriptions*

Object	Description
mplsLdpEntityStatsEntry	These entries augment the mplsLdpEntityTable by providing additional information for each entry.
mplsLdpAttemptedSessions	Not supported in this feature.
mplsLdpSesRejectedNoHelloErrors	A count of the session rejected/no hello error notification messages sent or received by this LDP entity.
mplsLdpSesRejectedAdErrors	A count of the session rejected/parameters advertisement mode error notification messages sent or received by this LDP entity.
mplsLdpSesRejectedMaxPduErrors	A count of the session rejected/parameters max PDU length error notification messages sent or received by this LDP entity.
mplsLdpSesRejectedLRErrors	A count of the session rejected/parameters label range notification messages sent or received by this LDP entity.
mplsLdpBadLdpIdentifierErrors	A count of the number of bad LDP identifier fatal errors detected by the session associated with this LDP entity.
mplsLdpBadPduLengthErrors	A count of the number of bad PDU length fatal errors detected by the session associated with this LDP entity.
mplsLdpBadMessageLengthErrors	A count of the number of bad message length fatal errors detected by the session associated with this LDP entity.
mplsLdpBadTlvLengthErrors	A count of the number of bad Type-Length-Value (TLV) length fatal errors detected by the session associated with this LDP entity.
mplsLdpMalformedTlvValueErrors	A count of the number of malformed TLV value fatal errors detected by the session associated with this LDP entity.
mplsLdpKeepAliveTimerExpErrors	A count of the number of session keepalive timer expired errors detected by the session associated with this LDP entity.
mplsLdpShutdownNotifReceived	A count of the number of shutdown notifications received related to the session associated with this LDP entity.

Object	Description
mplsLdpShutdownNotifSent	A count of the number of shutdown notifications sent related to the session associated with this LDP entity.

mplsLdpPeerTable

The table below lists the mplsLdpPeerTable objects and their descriptions.

Table 12 *mplsLdpPeerTable Objects and Descriptions*

Object	Description
mplsLdpPeerEntry	Information about a single peer that is related to a session (not accessible). Note This table is augmented by the mplsLdpSessionTable.
mplsLdpPeerLdpId	The LDP identifier of this LDP peer (not accessible) consists of the peer LSR ID (four octets) and the peer label space ID (two octets).
mplsLdpPeerLabelDistMethod	For any given LDP session, the method of label distribution. Values are downstreamOnDemand(1) and downstreamUnsolicited(2).
mplsLdpPeerLoopDetectionForPV	An indication of whether loop detection based on path vectors is disabled or enabled for this peer. For downstream unsolicited distribution (mplsLdpPeerLabelDistMethod is downstreamUnsolicited(2)), this object always has a value of disabled(0) and loop detection is disabled. For downstream-on-demand distribution (mplsLdpPeerLabelDistMethod is downstreamOnDemand(1)), this object has a value of enabled(1), provided that loop detection based on path vectors is enabled.
mplsLdpPeerPVL	If the value of mplsLdpPeerLoopDetectionForPV for this entry is enabled(1), this object represents that path vector limit for this peer. If the value of mplsLdpPeerLoopDetectionForPV for this entry is disabled(0), this value should be 0.

mplsLdpHelloAdjacencyTable

The table below lists the mplsLdpHelloAdjacencyTable objects and their descriptions.

Table 13 *mplsLdpHelloAdjacencyTable Objects and Descriptions*

Object	Description
mplsLdpHelloAdjacencyEntry	Each row represents a single LDP hello adjacency. An LDP session can have one or more hello adjacencies (not accessible).
mplsLdpHelloAdjIndex	An identifier for this specific adjacency (not accessible). The active hello adjacency has mplsLdpHelloAdjIndex equal to 1.
mplsLdpHelloAdjHoldTimeRem	The time remaining for this hello adjacency. This interval changes when the next hello message, which corresponds to this hello adjacency, is received.
mplsLdpHelloAdjType	This adjacency is the result of a link hello if the value of this object is link(1). Otherwise, this adjacency is a result of a targeted hello and its value is targeted(2).

mplsLdpSessionTable

The table below lists the mplsLdpSessionTable objects and their descriptions.

Table 14 *mplsLdpSessionTable Objects and Descriptions*

Object	Description
mplsLdpSessionEntry	An entry in this table represents information on a single session between an LDP entity and an LDP peer. The information contained in a row is read-only. This table augments the mplsLdpPeerTable.
mplsLdpSesState	The current state of the session. All of the states are based on the LDP or TDP state machine for session negotiation behavior. The states are as follows: <ul style="list-style-type: none"> • nonexistent(1) • initialized(2) • openrec(3) • opensent(4) • operational(5)
mplsLdpSesProtocolVersion	The version of the LDP protocol which this session is using. This is the version of the LDP protocol that has been negotiated during session initialization.

Object	Description
mplsLdpSesKeepAliveHoldTimeRem	The keepalive hold time remaining for this session.
mplsLdpSesMaxPduLen	The value of maximum allowable length for LDP PDUs for this session. This value could have been negotiated during the session initialization.
mplsLdpSesDiscontinuityTime	The value of sysUpTime on the most recent occasion when one or more of this session's counters suffered a discontinuity. The relevant counters are the specific instances of any Counter32 or Counter64 object contained in the mplsLdpSesStatsTable associated with this session. The initial value of this object is the value of sysUpTime when the entry was created in this table.

mplsLdpAtmSesTable

The table below lists the mplsLdpAtmSesTable objects and their descriptions.

Table 15 *mplsLdpAtmSesTable Objects and Descriptions*

Objects	Description
mplsLdpAtmSesEntry	An entry in this table represents information on a single label range intersection between an LDP entity and an LDP peer (not accessible).
mplsLdpAtmSesLRLowerBoundVpi	The minimum VPI number for this range (not accessible).
mplsLdpAtmSesLRLowerBoundVci	The minimum VCI number for this range (not accessible).
mplsLdpAtmSesLRUpperBoundVpi	The maximum VPI number for this range (read-only).
mplsLdpAtmSesLRUpperBoundVci	The maximum VCI number for this range (read-only).

mplsLdpSesStatsTable

The table below lists the mplsLdpSesStatsTable objects and their descriptions.

Table 16 *mplsLdpSesStatsTable Objects and Descriptions*

Object	Description
mplsLdpSesStatsEntry	An entry in this table represents statistical information on a single session between an LDP entity and an LDP peer. This table augments the mplsLdpPeerTable.
mplsLdpSesStatsUnkMesTypeErrors	This object is the count of the number of unknown message type errors detected during this session.
mplsLdpSesStatsUnkTlvErrors	This object is the count of the number of unknown TLV errors detected during this session.

VPN Contexts in MPLS LDP MIB Version 8 Upgrade

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 LDP contexts. Each context is independent from all others and contains data specific only to that context.

This feature adds support for different contexts for different MPLS VPNs. Users of the MIB can view MPLS LDP processes for a given MPLS VPN. The VPN Aware LDP MIB feature does not change the syntax of the IETF MPLS-LDP MIB. It changes the number and types of entries within the tables.

The IETF MPLS-LDP MIB can show information about only one context at a time. You can specify a context, either a global context or an MPLS VPN context, using an SNMP security name.

The following sections describe topics related to the VPN Aware LDP MIB feature:

SNMP Context

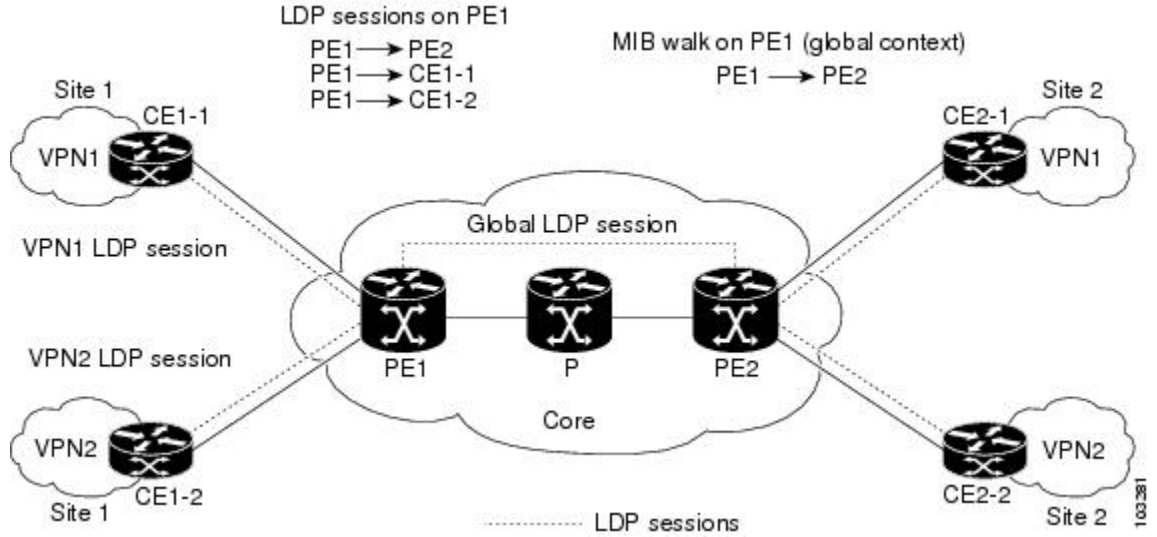
SNMP contexts provide VPN users with a secure way of accessing MIB data. When a VPN is associated with a context, that VPN's specific MIB data exists in that context. Associating a VPN with a context enables service providers to manage networks with multiple VPNs. Creating and associating a context with a VPN enables a provider to prevent the users of one VPN from accessing information about users of other VPNs on the same networking device.

VPN Aware LDP MIB Sessions

Before the VPN Aware LDP MIB features, an SNMP query to the MPLS LDP MIB returned information about global sessions only. A query did not return information about LDP sessions in a VPN context. The IETF MPLS LDP MIB retrieved information from global routing tables, but did not retrieve information from VPN routing and forwarding instances (VRFs) that store per-VPN routing data. The MPLS LDP MIB looked only at LDP processes in the global context and ignored all other sessions. A query on a VRF returned no information. You can view LDP processes in a VPN context.

The figure below shows a sample MPLS VPN network with the MPLS LDP sessions prior to the implementation of the VPN Aware LDP MIB feature.

Figure 7 MPLS LDP Sessions Setup Before VPN Aware LDP MIB Feature



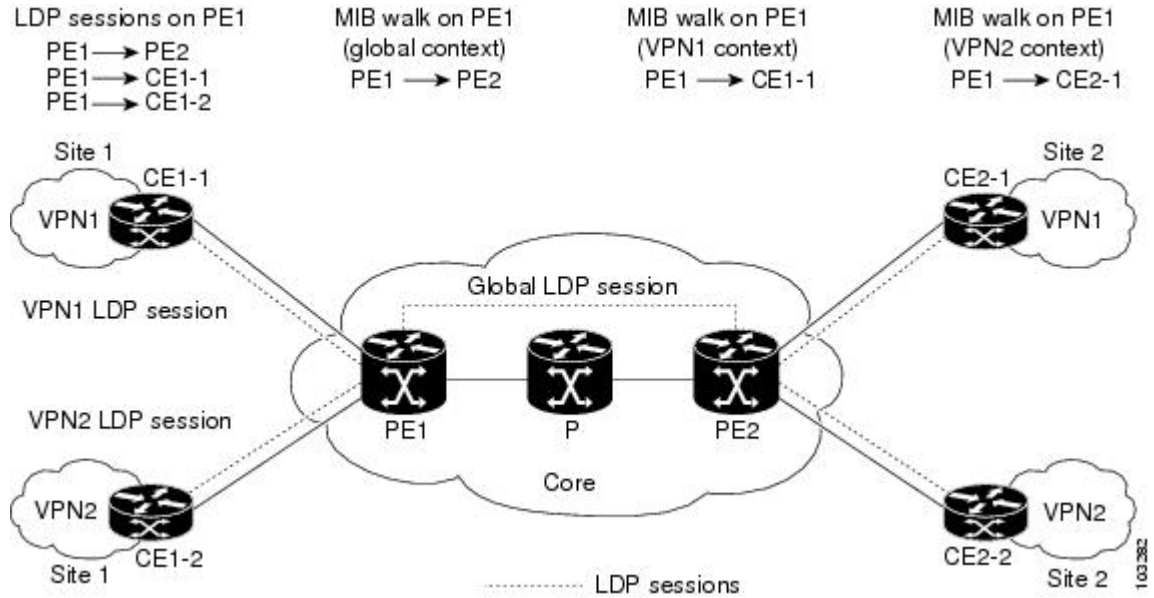
A MIB walk prior to this software release displayed only global session information.

With the VPN Aware LDP MIB enhancement, an SNMP query to the IETF MPLS-LDP-MIB supports both global and VPN contexts. This feature allows you to enter LDP queries on any VRF and on the core (global context). A query can differentiate between LDP sessions from different VPNs. LDP session information for a VPN stays in the context of that VPN. Therefore, the information from one VPN is not available to a user of a different VPN. The VPN Aware update to the LDP MIB also allows you to view LDP processes operating in a Carrier Supporting Carrier (CSC) network.

In an MPLS VPN, a service provider edge router (PE) might contain VRFs for several VPNs as well as a global routing table. To set up separate LDP processes for different VPNs on the same device, you need to configure each VPN with a unique securityName, contextName, and View-based Access Control Model (VACM) view. The VPN securityName must be configured for the IETF MPLS LDP MIB.

The figure below shows LDP sessions for a sample MPLS VPN network with the VPN Aware LDP MIB feature.

Figure 8 MPLS LDP Sessions with the VPN Aware LDP MIB Feature



With the VPN Aware LDP MIB feature, you can do MIB queries or MIB walks for an MPLS VPN LDP session or a global LDP session.



Note

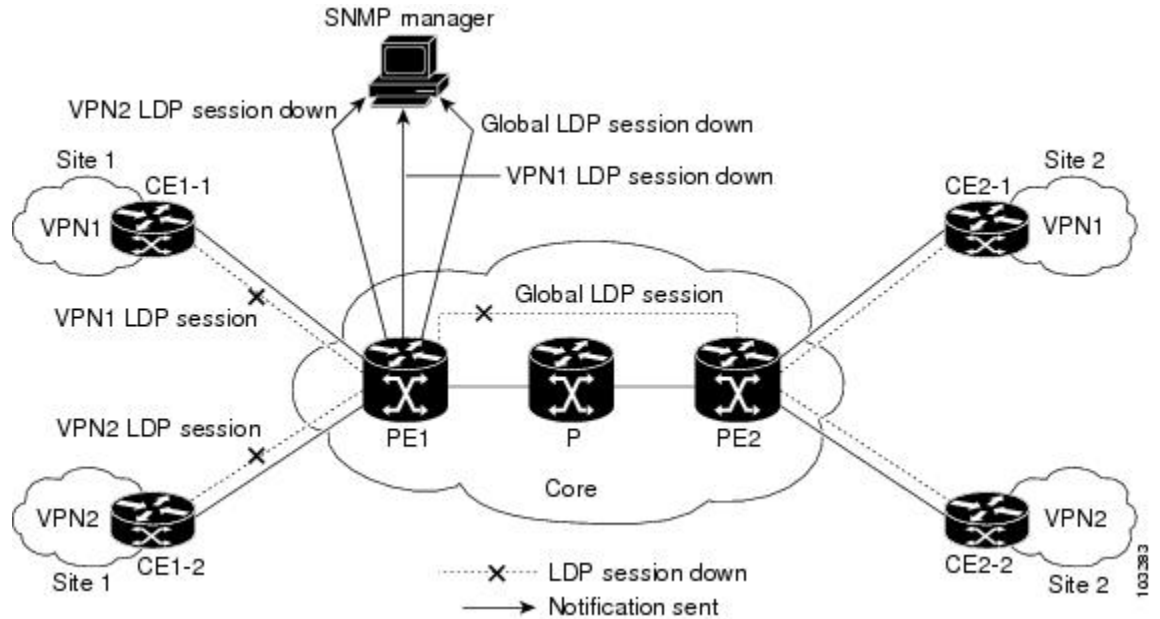
To verify LDP session information for a specific VPN, use the **show mpls ldp neighbor vrf vpn-name detail** command.

VPN Aware LDP MIB Notifications

Before the VPN Aware LDP MIB feature, all notification messages for MPLS LDP sessions were sent to the same designated network management station (NMS) in the network. The notifications were enabled with the **snmp-server enable traps mpls ldp** command.

The figure below shows LDP notifications that were sent before the implementation of the VPN Aware LDP MIB feature.

Figure 9 LDP Notifications Sent Before the VPN Aware LDP MIB Feature



The VPN Aware LDP MIB feature supports LDP notifications for multiple LDP contexts for VPNs. LDP notifications can be generated for the core (global context) and for different VPNs. You can cause notifications be sent to different NMS hosts for different LDP contexts. LDP notifications associated with a specific VRF are sent to the NMS designated for that VRF. LDP global notifications are sent to the NMS configured to receive global traps.

To enable LDP context notifications for the VPN Aware LDP MIB feature, use either the SNMP object `mplsLdpSessionsUpDownEnable` (in the global LDP context only) or the following extended global configuration commands.

To enable LDP notifications for the global context, use the following commands on a PE router:

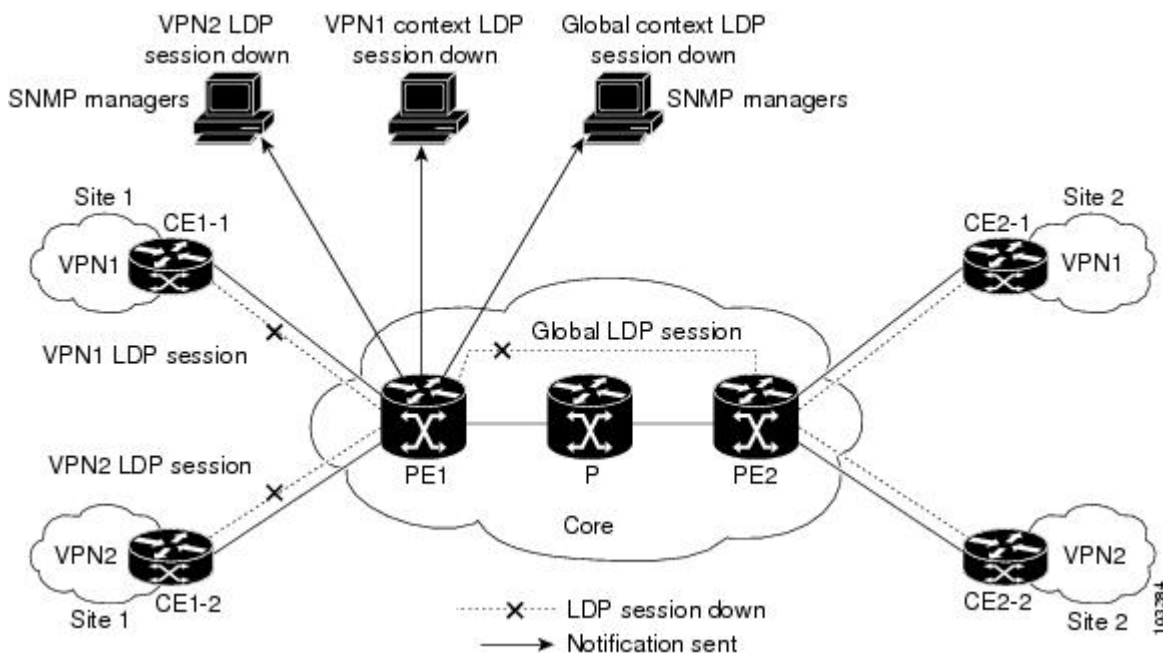
```
Router(config)# snmp-server host host-address traps community mpls-ldp
Router(config)# snmp-server enable traps mpls ldp
```

To enable LDP notifications for a VPN context, use the following commands on a PE router:

```
Router(config)# snmp-server host host-address vrf vrf-name version {v1|v2c|v3}
community community-string udp-port upd-port mpls-ldp
Router(config)# snmp-server enable traps mpls ldp
```

The figure below shows LDP notifications with the VPN Aware LDP MIB feature.

Figure 10 LDP Notifications With the VPN Aware LDP MIB Feature



How to Configure MPLS LDP MIB Version 8 Upgrade

- [Enabling the SNMP Agent, page 8](#)
- [Enabling Distributed Cisco Express Forwarding, page 60](#)
- [Enabling MPLS Globally, page 61](#)
- [Enabling LDP Globally, page 62](#)
- [Enabling MPLS on an Interface, page 62](#)
- [Enabling LDP on an Interface, page 63](#)
- [Configuring a VPN Aware LDP MIB, page 64](#)
- [Verifying MPLS LDP MIB Version 8 Upgrade, page 70](#)

Enabling the SNMP Agent

SUMMARY STEPS

1. enable
2. show running-config
3. configure terminal
4. snmp-server community *string* [view *view-name*] [ro *number*]
5. end
6. write memory
7. show running-config

DETAILED STEPS

Command or Action	Purpose
Step 1 enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2 show running-config Example: Router# show running-config	Displays the running configuration of the router so that you can determine if an SNMP agent is already running on the device. 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 desired.
Step 3 configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 4 snmp-server community <i>string</i> [view <i>view-name</i>] [ro <i>number</i>] Example: Router(config)# snmp-server community public ro	Configures read-only (ro) community strings for the MPLS Label Distribution Protocol (LDP) MIB. <ul style="list-style-type: none"> • The <i>string</i> argument functions like a password, permitting access to SNMP functionality on label switch routers (LSRs) in an MPLS network. • The optional ro keyword configures read-only (ro) access to the objects in the MPLS LDP MIB.
Step 5 end Example: Router(config)# end	Exits to privileged EXEC mode.

Command or Action	Purpose
Step 6 write memory Example: Router# write memory	Writes the modified SNMP configuration into NVRAM of the router, permanently saving the SNMP settings.
Step 7 show running-config Example: Router# show running-config	Displays the running configuration of the router so that you can determine if an SNMP agent is already running on the device. If you see any snmp-server statements, SNMP has been enabled on the router. If any SNMP information is displayed, you can modify the information or change it as desired.

Enabling Distributed Cisco Express Forwarding

Perform this task to enable Cisco Express Forwarding or distributed Cisco Express Forwarding.

SUMMARY STEPS

1. enable
2. configure terminal
3. ip cef distributed
4. end

DETAILED STEPS

Command or Action	Purpose
Step 1 enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2 configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3 ip cef distributed Example: Router(config)# ip cef distributed	Enables distributed Cisco Express Forwarding.

	Command or Action	Purpose
Step 4	end Example: Router(config)# end	Exits to privileged EXEC mode.

Enabling MPLS Globally

Perform this task to enable MPLS globally.

SUMMARY STEPS

1. enable
2. configure terminal
3. mpls ip
4. end

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	mpls ip Example: Router(config)# mpls ip	Enables MPLS forwarding of IPv4 packets along normally routed paths for the platform.
Step 4	end Example: Router(config)# end	Exits to privileged EXEC mode.

Enabling LDP Globally

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol {ldp | tdp}**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	mpls label protocol {ldp tdp} Example: Router(config)# mpls label protocol ldp	Specifies the platform default label distribution protocol. TDP might not be supported in all software releases.
Step 4	end Example: Router(config)# end	Exits to privileged EXEC mode.

Enabling MPLS on an Interface

Perform this task to enable MPLS on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/subslot/port* [*.subinterface-number*]
4. **mpls ip**
5. **end**

DETAILED STEPS

Command or Action	Purpose
Step 1 enable Example: <pre>Router> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2 configure terminal Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3 interface <i>type slot/subslot/port</i> [<i>.subinterface-number</i>] Example: <pre>Router(config)# interface FastEthernet 1/0/0</pre>	Configures an interface type and enters interface configuration mode.
Step 4 mpls ip Example: <pre>Router(config-if)# mpls ip</pre>	Enables MPLS forwarding of IPv4 packets along normally routed paths for a particular interface.
Step 5 end Example: <pre>Router(config-if)# end</pre>	Exits to privileged EXEC mode.

Enabling LDP on an Interface

Perform this task to enable LDP on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot / subslot / port* [*. subinterface-number*]
4. **mpls label protocol ldp**
5. **end**

DETAILED STEPS

Command or Action	Purpose
Step 1 enable Example: <pre>Router> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2 configure terminal Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3 interface <i>type slot / subslot / port</i> [<i>. subinterface-number</i>] Example: <pre>Router(config)# interface FastEthernet 1/0/0</pre>	Configures an interface type and enters interface configuration mode.
Step 4 mpls label protocol ldp Example: <pre>Router(config-if)# mpls label protocol ldp</pre>	Specifies the label distribution protocol to be used on a given interface.
Step 5 end Example: <pre>Router(config-if)# end</pre>	Exits to privileged EXEC mode.

Configuring a VPN Aware LDP MIB

- [Configuring SNMP Support for a VPN, page 65](#)
- [Configuring an SNMP Context for a VPN, page 66](#)
- [Associating an SNMP VPN Context with SNMPv1 or SNMPv2, page 68](#)

Configuring SNMP Support for a VPN

Perform this task to configure SNMP support for a Virtual Private Network (VPN) or a remote VPN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server host** *host-address* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] [**vrf** *vrf-name*]
4. **snmp-server engineID remote** *ip-address* [**udp-port** *udp-port-number*] [**vrf** *vrf-name*] *engineid-string*
5. **end**

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 enable</p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
<p>Step 2 configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 snmp-server host <i>host-address</i> [traps informs] [version {1 2c 3 [auth noauth priv]}] <i>community-string</i> [udp-port <i>port</i>] [<i>notification-type</i>] [vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>Router(config)# snmp-server host example.com vrf trap-vrf</pre>	<p>Specifies the recipient of an SNMP notification operation and specifies the Virtual Private Network (VPN) routing and forwarding (VRF) instance table to be used for the sending of SNMP notifications.</p>
<p>Step 4 snmp-server engineID remote <i>ip-address</i> [udp-port <i>udp-port-number</i>] [vrf <i>vrf-name</i>] <i>engineid-string</i></p> <p>Example:</p> <pre>Router(config)# snmp-server engineID remote 172.16.20.3 vrf traps-vrf 80000009030000B064EFE100</pre>	<p>Configures a name for the remote SNMP engine on a router.</p>

Command or Action	Purpose
Step 5 <code>end</code> Example: <code>Router(config)# end</code>	Exits to privileged EXEC mode.

Configuring an SNMP Context for a VPN

Perform this task to configure an SNMP context for a VPN. This sets up a unique SNMP context for a VPN, which allows you to access the VPN's LDP session information.

- [SNMP Context, page 54](#)
- [VPN Route Distinguishers, page 66](#)

SNMP Context

SNMP contexts provide VPN users with a secure way of accessing MIB data. When a VPN is associated with a context, that VPN's specific MIB data exists in that context. Associating a VPN with a context enables service providers to manage networks with multiple VPNs. Creating and associating a context with a VPN enables a provider to prevent the users of one VPN from accessing information about users of other VPNs on the same networking device.

VPN Route Distinguishers

A route distinguisher (RD) creates routing and forwarding tables for a VPN. Cisco software adds the RD to the beginning of the customer's IPv4 prefixes to change them into globally unique VPN-IPv4 prefixes.

Either the RD is an autonomous system number (ASN)-relative RD, in which case it is composed of an autonomous system number and an arbitrary number, or it is an IP-address-relative RD, in which case it is composed of an IP address and an arbitrary number. You can enter an RD in either of these formats:

- 16-bit ASN: your 32-bit number, for example, 101:3.
- 32-bit IP address: your 16-bit number, for example, 192.168.122.15:1.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server context** *context-name*
4. **ip vrf** *vrf-name*
5. **rd** *route-distinguisher*
6. **context** *context-name*
7. **route-target** [**import** | **export** | **both**] *route-target-ext-community*
8. **end**

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 <code>enable</code></p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
<p>Step 2 <code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 <code>snmp-server context context-name</code></p> <p>Example:</p> <pre>Router(config)# snmp-server context context1</pre>	<p>Creates and names an SNMP context.</p>
<p>Step 4 <code>ip vrf vrf-name</code></p> <p>Example:</p> <pre>Router(config)# ip vrf vrf1</pre>	<p>Configures a Virtual Private Network (VPN) routing and forwarding instance (VRF) table and enters VRF configuration mode.</p>
<p>Step 5 <code>rd route-distinguisher</code></p> <p>Example:</p> <pre>Router(config-vrf)# rd 100:120</pre>	<p>Creates a VPN route distinguisher.</p>
<p>Step 6 <code>context context-name</code></p> <p>Example:</p> <pre>Router(config-vrf)# context context1</pre>	<p>Associates an SNMP context with a particular VRF.</p>
<p>Step 7 <code>route-target [import export both] route-target-ext-community</code></p> <p>Example:</p> <pre>Router(config-vrf)# route-target export 100:1000</pre>	<p>(Optional) Creates a route-target extended community for a VRF.</p>

Command or Action	Purpose
Step 8 end Example: Router(config)# end	Exits to privileged EXEC mode.

Associating an SNMP VPN Context with SNMPv1 or SNMPv2

Perform this task to associate an SNMP VPN context with SNMPv1 or SNMPv2. This allows you to access LDP session information for a VPN using SNMPv1 or SNMPv2.

SNMPv1 or SNMPv2 Security: SNMPv1 and SNMPv2 are not as secure as SNMPv3. SNMP Versions 1 and 2 use plain text communities and do not perform the authentication or security checks that SNMP Version 3 performs.

To configure the VPN Aware LDP MIB feature when using SNMP Version 1 or SNMP Version 2, you need to associate a community name with a VPN. This association causes SNMP to process requests coming in for a particular community string only if they come in from the configured VRF. If the community string contained in the incoming packet does not have an associated VRF, the packet is processed only if it came in through a non-VRF interface. This process prevents users outside the VPN from using a clear text community string to query the VPN data. However, this is not as secure as using SNMPv3.

SUMMARY STEPS

1. enable
2. configure terminal
3. **snmp-server user** *username group-name* [**remote host** [**udp-port port**]] {**v1** | **v2c** | **v3** [**encrypted**] [**auth** {**md5** | **sha**} *auth-password*]} [**access access-list**]
4. **snmp-server group** *group-name* {**v1** | **v2c** | **v3**{**auth** | **noauth** | **priv**}} [**context context-name**] [**read readview**] [**write writeview**] [**notify notifyview**] [**access access-list**]
5. **snmp-server view** *view-name oid-tree* {**included** | **excluded**}
6. **snmp-server enable traps** [*notification-type*]
7. **snmp-server host** *host-address* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] [*community-string*] [**udp-port port**] [**notification-type**] [**vrf vrf-name**]
8. **snmp mib community-map** *community-name* [**context context-name**] [**engineid engine-id**] [**security-name security-name**] **target-list** *vpn-list-name*
9. **snmp mib target list** *vpn-list-name* {**vrf vrf-name** | **host ip-address**}
10. no **snmp-server trap authentication vrf**
11. exit

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
Step 3	<p>snmp-server user <i>username group-name</i> [remote host [udp-port <i>port</i>]] {v1 v2c v3 [encrypted] [auth {md5 sha} <i>auth-password</i>]} [access <i>access-list</i>]</p> <p>Example:</p> <pre>Router(config)# snmp-server user customer1 group1 v1</pre>	<p>Configures a new user to an SNMP group.</p>
Step 4	<p>snmp-server group <i>group-name</i> {v1 v2c v3{auth noauth priv}} [context <i>context-name</i>] [read <i>readview</i>] [write <i>writeview</i>] [notify <i>notifyview</i>] [access <i>access-list</i>]</p> <p>Example:</p> <pre>Router(config)# snmp-server group group1 v1 context context1 read view1 write view1 notify view1</pre>	<p>Configures a new SNMP group or a table that maps SNMP users to SNMP views.</p> <ul style="list-style-type: none"> Use the context <i>context-name</i> keyword and argument to associate the specified SNMP group with a configured SNMP context.
Step 5	<p>snmp-server view <i>view-name oid-tree</i> {included excluded}</p> <p>Example:</p> <pre>Router(config)# snmp-server view view1 ipForward included</pre>	<p>Creates or updates a view entry.</p>
Step 6	<p>snmp-server enable traps [<i>notification-type</i>]</p> <p>Example:</p> <pre>Router(config)# snmp-server enable traps</pre>	<p>Enables all SNMP notifications (traps or informs) available on your system.</p>

Command or Action	Purpose
<p>Step 7 <code>snmp-server host <i>host-address</i> [traps informs] [version { 1 2c 3 [auth noauth priv] }] <i>community-string</i> [udp-port <i>port</i>] [notification-type] [vrf <i>vrf-name</i>]</code></p> <p>Example:</p> <pre>Router(config)# snmp-server host 10.0.0.1 vrf customer1 public udp-port 7002</pre>	Specifies the recipient of an SNMP notification operation.
<p>Step 8 <code>snmp mib community-map <i>community-name</i> [context <i>context-name</i>] [engineid <i>engine-id</i>] [security-name <i>security-name</i>] target-list <i>vpn-list-name</i></code></p> <p>Example:</p> <pre>Router(config)# snmp mib community-maps community1 context context1 target-list commAVpn</pre>	Associates an SNMP community with an SNMP context, Engine ID, or security name.
<p>Step 9 <code>snmp mib target list <i>vpn-list-name</i> {vrf <i>vrf-name</i> host <i>ip-address</i>}</code></p> <p>Example:</p> <pre>Router(config)# snmp mib target list commAVpn vrf vrf1</pre>	Creates a list of target VRFs and hosts to associate with an SNMP community.
<p>Step 10 <code>no snmp-server trap authentication vrf</code></p> <p>Example:</p> <pre>Router(config)# no snmp-server trap authentication vrf</pre>	<p>(Optional) Disables all SNMP authentication notifications (traps and informs) generated for packets received on VRF interfaces.</p> <ul style="list-style-type: none"> Use this command to disable authentication traps only for those packets on VRF interfaces with incorrect community associations.
<p>Step 11 <code>exit</code></p> <p>Example:</p> <pre>Router(config) exit</pre>	Exits to privileged EXEC mode.

Verifying MPLS LDP MIB Version 8 Upgrade

Perform a MIB walk using your SNMP management tool to verify that the MPLS LDP MIB Version 8 Upgrade feature is functioning.

Configuration Examples for MPLS LDP MIB Version 8 Upgrade

- [MPLS LDP MIB Version 8 Upgrade Examples, page 71](#)
- [Configuring a VPN Aware SNMP Context for SNMPv1 or SNMPv2 Example, page 71](#)

MPLS LDP MIB Version 8 Upgrade Examples

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

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
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 that have 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 LDP globally and then on an interface:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# mpls label protocol ldp
Router(config)# interface FastEthernet1/0/0
Router(config-if)# mpls label protocol ldp
Router(config-if)# end
```

Configuring a VPN Aware SNMP Context for SNMPv1 or SNMPv2 Example

The following configuration example shows how to configure a VPN Aware SNMP context for the MPLS LDP MIB Version 8 with SNMPv1 or SNMPv2:

```
snmp-server context A
snmp-server context B
ip vrf CustomerA
  rd 100:110
  context A
  route-target export 100:1000
  route-target import 100:1000
!
ip vrf CustomerB
  rd 100:120
  context B
  route-target export 100:2000
  route-target import 100:2000
!
interface FastEthernet0/3/1
  description Belongs to VPN A
```

```

ip vrf forwarding CustomerA
ip address 10.0.0.0 255.255.0.0

interface FastEthernet0/3/2
description Belongs to VPN B
ip vrf forwarding CustomerB
ip address 10.0.0.1 255.255.0.0
snmp-server user commA grp1A v1
snmp-server user commA grp2A v2c
snmp-server user commB grp1B v1
snmp-server user commB grp2B v2c
snmp-server group grp1A v1 context A read viewA write viewA notify viewA
snmp-server group grp1B v1 context B read viewB write viewB notify viewB
snmp-server view viewA ipForward included
snmp-server view viewA ciscoPingMIB included
snmp-server view viewB ipForward included
snmp-server view viewB ciscoPingMIB included
snmp-server enable traps
snmp-server host 10.0.0.3 vrf CustomerA commA udp-port 7002
snmp-server host 10.0.0.4 vrf CustomerB commB udp-port 7002
snmp mib community-map commA context A target-list commAvpn
! Configures source address validation
snmp mib community-map commB context B target-list commBvpn
! Configures source address validation
snmp mib target list commAvpn vrf CustomerA
! Configures a list of VRFs or from which community commA is valid
snmp mib target list commBvpn vrf CustomerB
! Configures a list of VRFs or from which community commB is valid

```

Additional References

Related Documents

Related Topic	Document Title
MPLS LDP configuration tasks	MPLS Label Distribution Protocol (LDP)
A description of SNMP agent support for the MPLS Traffic Engineering MIB (MPLS TE MIB)	MPLS Traffic Engineering (TE) MIB
A description of MPLS differentiated types of service across an MPLS network	MPLS Quality of Service
SNMP commands	<i>Network Management Command Reference</i>
SNMP configuration	Configuring SNMP Support
SNMP Support for VPNs	

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
<ul style="list-style-type: none"> • MPLS Label Distribution Protocol MIB (draft-ietf-mpls-ldp-mib-08.txt) • SNMP-VACM-MIB The View-based Access Control Model (ACM) MIB for SNMP 	<p>To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL:</p> <p>http://www.cisco.com/go/mibs</p>

RFCs

RFCs	Title
<p>RFC 2233</p> <p>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.</p>	<p><i>Interfaces MIB</i></p>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>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.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	<p>http://www.cisco.com/techsupport</p>

Feature Information for MPLS LDP MIB Version 8 Upgrade

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 17 **Feature Information for MPLS LDP MIB Version 8 Upgrade**

Feature Name	Releases	Feature Information
MPLS Label Distribution Protocol MIB	12.0(11)ST	The MPLS Label Distribution Protocol (LDP) MIB Version 8 Upgrade feature enhances the LDP MIB to support the Internet Engineering Task Force (IETF) draft Version 8. In Cisco IOS Release 12.0(11)ST, this feature was introduced to provide SNMP agent support for the MPLS LDP MIB on the Cisco 7200, Cisco 7500, and Cisco 12000 series routers. In Cisco IOS Release 12.2(2)T, this feature was added to this release to provide SNMP agent support for the MPLS LDP MIB on Cisco 7200 and Cisco 7500 series routers. In Cisco IOS Release 12.0(21)ST, this feature was added to this release to provide SNMP agent and LDP notification support for the MPLS LDP MIB on Cisco 7200, Cisco 7500, and Cisco 12000 series Internet routers. In Cisco IOS Release 12.0(22)S, Version 1 was integrated into Cisco IOS Release 12.0(22)S. In Cisco IOS Release 12.0(24)S, this feature was upgraded to Version 8 in Cisco IOS Release 12.0(24)S. In Cisco IOS Release 12.0(27)S, support for the MPLS VPN-VPN Aware LDP MIB feature was added. This feature was integrated into Cisco IOS Release 12.2(18)S. This feature was integrated into Cisco IOS Release 12.2(33)SRA. This feature was integrated into Cisco IOS Release 12.2(33)SXH. In Cisco IOS Release 12.2(33)SRB, this MIB was
	12.2(2)T	
	12.0(21)ST	
	12.0(22)S	
	12.0(24)S	
	12.0(27)S	
	12.2(18)S	
	12.2(33)SRA	
	12.2(33)SXH	
	12.2(33)SRB	
	12.2(33)SB)	
	Cisco IOS XE Release 2.1	

Feature Name	Releases	Feature Information
		<p>deprecated and replaced by MPLS-LDP-STD-MIB (RVC 3815).</p>
		<p>In Cisco IOS Release 12.2(33)SB, this MIB was deprecated and replaced by MPLS-LDP-STD-MIB (RVC 3815).</p>
		<p>This feature was integrated into Cisco IOS XE Release 2.1 and implemented on Cisco ASR 1000 Series Aggregation Services Routers.</p>
		<p>The following commands were introduced or modified: context, show mpls ldp neighbor, snmp mib community-map, snmp mib target list, snmp-server community, snmp-server context, snmp-server enable traps (MPLS), snmp-server group, snmp-server host, snmp-server trap authentication vrf.</p>

Glossary

ATM -- Asynchronous Transfer Mode. The international standard for cell relay in which multiple service types (such as voice, video, or data) are conveyed in fixed-length (53-byte) cells. Fixed-length cells allow cell processing to occur in hardware, thereby reducing transit delays. ATM is designed to take advantage of high-speed transmission media, such as E3, SONET, and T3 .

downstream-on-demand distribution--A label distribution method in which a downstream label switch router (LSR) sends a binding upstream only if the upstream LSR requests it.

downstream unsolicited distribution--A label distribution method in which labels are dispersed if a downstream label switch router (LSR) needs to establish a new binding with its neighboring upstream LSR. For example, an edge LSR might enable a new interface with another subnet. The LSR then announces to the upstream router a binding to reach this network.

informs --A type of notification message that is more reliable than a conventional trap notification message, because the informs message notification requires acknowledgment, but a trap notification does not.

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

label distribution--The techniques and processes that are used by label switch routers (LSRs) to exchange label binding information for supporting hop-by-hop forwarding along normally routed paths.

LDP --Label Distribution Protocol. The protocol that supports Multiprotocol Label Switching (MPLS) hop-by-hop forwarding and the distribution of bindings between labels and network prefixes.

LSP --label switched path. A configured connection between two label switch routers (LSRs) in which label-switching techniques are used for packet forwarding; also a specific path through an Multiprotocol Label Switching (MPLS) network.

LSR --label switch router. A Multiprotocol Label Switching (MPLS) node that can forward native Layer 3 packets. The LSR forwards a packet based on the value of a label attached to the packet.

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

MPLS --Multiprotocol Label Switching. A switching method for the forwarding of IP traffic through the use of a label. This label instructs the routers and the switches in the network where to forward the packets based on preestablished IP routing information.

MPLS label distribution--A constraint-based routing algorithm for routing label-switched path (LSP) tunnels.

NMS --network management station. 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. In the context of Simple Network Management Protocol (SNMP), an NMS is a device that performs SNMP queries to the SNMP agent of a managed device to retrieve or modify information.

notification --A message sent by a Simple Network Management Protocol (SNMP) agent to a network management station, console, or terminal to indicate that a significant network event has occurred. See also trap.

RSVP --Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network. Applications running on IP end systems can use RSVP to indicate to other nodes the nature of the packet streams they want to receive by specifying such items as bandwidth, jitter, and maximum burst.

RTR --Response Time Reporter. A tool that allows you to monitor network performance, network resources, and applications by measuring response times and availability.

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

SNMP communities--Authentication scheme that enables an intelligent network device to validate SNMP requests.

SNMPv2c --Version 2c of the Simple Network Management Protocol. SNMPv2c supports centralized as well as distributed network management strategies and includes improvements in the Structure of Management Information (SMI), protocol operations, management architecture, and security.

SNMPv3 --Version 3 of the Simple Network Management Protocol. Interoperable standards-based protocol for network management. SNMPv3 provides secure access to devices by a combination of authenticating and encrypting packets over the network.

TLV --Type-Length-Value. A mechanism used by several routing protocols to carry a variety of attributes. Cisco Discovery Protocol (CDP), Label Discovery Protocol (LDP), and Border Gateway Protocol (BGP) are examples of protocols that use TLVs. BGP uses TLVs to carry attributes such as Network Layer Reachability Information (NLRI), Multiple Exit Discriminator (MED), and local preference.

trap --A message sent by a Simple Network Management Protocol (SNMP) agent to a network management station, console, or terminal to indicate that a significant network event 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. See also notification.

VCC --virtual channel connection. A logical circuit, made up of virtual channel links (VCLs), that carries data between two endpoints in an ATM network. Sometimes called a virtual circuit connection.

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

VCL --virtual channel link. 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 virtual channel identifier (VCI), is used to identify the next network virtual channel link (VCL) as the cell passes through a series of ATM switches on its way to its final destination.

VPN --Virtual Private Network. A network that enables IP traffic to use tunneling to travel securely over a public TCP/IP network.

VRF --VPN routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

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



MPLS Traffic Engineering MIB

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

- [Finding Feature Information, page 79](#)
- [Restrictions for the MPLS Traffic Engineering MIB, page 79](#)
- [Information About the MPLS Traffic Engineering MIB, page 80](#)
- [How to Configure the MPLS Traffic Engineering MIB, page 88](#)
- [Configuration Examples for the MPLS Traffic Engineering MIB, page 90](#)
- [Additional References, page 91](#)
- [Feature Information for the MPLS Traffic Engineering MIB, page 92](#)
- [Glossary, page 93](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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.

Restrictions for the MPLS Traffic Engineering MIB

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

Information About the MPLS Traffic Engineering MIB

- [MPLS Traffic Engineering MIB Cisco Implementation, page 80](#)
- [Capabilities Supported by the MPLS Traffic Engineering MIB, page 80](#)
- [Notification Generation Events, page 81](#)
- [Notification Implementation, page 81](#)
- [Benefits of the MPLS Traffic Engineering MIB, page 82](#)
- [MPLS Traffic Engineering MIB Layer Structure, page 82](#)
- [Features and Technologies Related to MPLS Traffic Engineering MIB, page 82](#)
- [Supported Objects in the MPLS Traffic Engineering MIB, page 82](#)
- [CLI Access to MPLS Traffic Engineering MIB Information, page 87](#)

MPLS Traffic Engineering MIB Cisco Implementation

The MPLS TE MIB is based on the Internet Engineering Task Force (IETF) draft MIB entitled *draft-ietf-mpls-te-mib-05.txt* which includes objects describing features that support MPLS TE.

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

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

- [MPLS Traffic Engineering Overview, page 80](#)

MPLS Traffic Engineering Overview

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

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

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

Capabilities Supported by the MPLS Traffic Engineering MIB

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

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

Notification Generation Events

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

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

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

An `mplsTunnelRerouted` notification is sent to the NMS under the following conditions:

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

The `mplsTunnelReoptimized` notification is not generated when an MPLS traffic engineering tunnel is reoptimized. However, an `mplsTunnelReroute` notification is generated. Thus, at the NMS, you cannot distinguish between a tunnel reoptimization event and tunnel reroute event.

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

Notification Implementation

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

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

Benefits of the MPLS Traffic Engineering MIB

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

MPLS Traffic Engineering MIB Layer Structure

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

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

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

Features and Technologies Related to MPLS Traffic Engineering MIB

The MPLS TE MIB feature is used with the following features and technologies:

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

Supported Objects in the MPLS Traffic Engineering MIB

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

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

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

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

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

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

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

Following is a list and description of each entry.

- **mplsTunnelIndex**--Same as tunnel ID (c).
- **mplsTunnelInstance**--Tunnel instance of the LSP; 0 for head configurations (b).
- **mplsTunnelIngressLSRId**--Source IP address of the LSP; 0 for head configurations (b).
- **mplsTunnelEgressLSRId**--Destination IP address of the tunnel (c).
- **mplsTunnelName**--Command name for the tunnel interfaces (a).
- **mplsTunnelDescr**--Descriptive name for tunnel configurations and LSPs (c).
- **mplsTunnelIsIf**--Indicator of whether the entry represents an interface (c).
- **mplsTunnelIfIndex**--Index of the tunnel interface within the ifMIB (a).
- **mplsTunnelXCPointer**--(For midpoints only - no tails) Pointer for the LSP within the **mplsXCTable** of the MPLS LSR MIB (b).
- **mplsTunnelSignallingProto**--Signaling protocol used by tunnels (c).
- **mplsTunnelSetupPrio**--Setup priority of the tunnel (c).
- **mplsTunnelHoldingPrio**--Holding priority of the tunnel (c).
- **mplsTunnelSessionAttributes**--Session attributes (c).
- **mplsTunnelOwner**--Tunnel owner (c).
- **mplsTunnelLocalProtectInUse**--Not implemented (c).
- **mplsTunnelResourcePointer**--Pointer into the Resource Table (b).
- **mplsTunnelInstancePriority**--Not implemented (b).
- **mplsTunnelHopTableIndex**--Index into the Hop Table (a).
- **mplsTunnelARHopTableIndex**--Index into the AR Hop Table (b).
- **mplsTunnelCHopTableIndex**--Index into the C Hop Table (b).

- `mplsTunnelPrimaryTimeUp`--Amount of time, in seconds, that the current path has been up (a).
- `mplsTunnelPathChanges`--Number of times a tunnel has been resignalled (a).
- `mplsTunnelLastPathChange`--Amount of time, in seconds, since the last path resignaling occurred (a).
- `mplsTunnelCreationTime`--Time stamp when the tunnel was created (a).
- `mplsTunnelStateTransitions`--Number of times the tunnel has changed state (a).
- `mplsTunnelIncludeAnyAffinity`--Not implemented (a).
- `mplsTunnelIncludeAllAffinity`--Attribute bits that must be set for the tunnel to traverse a link (a).
- `mplsTunnelExcludeAllAffinity`--Attribute bits that must *not* be set for the tunnel to traverse a link (a).
- `mplsTunnelPathInUse`--Path option number being used for the tunnel's path. If no path option is active, this object will be 0 (a).
- `mplsTunnelRole`--Role of the tunnel on the router; that is, head, midpoint, or tail (c).
- `mplsTunnelTotalUptime`--Amount of time, in seconds, that the tunnel has been operationally up (a).
- `mplsTunnelInstanceUptime`--Not implemented (b).
- `mplsTunnelAdminStatus`--Administrative status of a tunnel (c).
- `mplsTunnelOperStatus`--Actual operating status of a tunnel (c).
- `mplsTunnelRowStatus`--This object is used in conjunction with configuring a new tunnel. This object will always be seen as "active" (a).
- `mplsTunnelStorageType`--Storage type of a tunnel entry (c).
- `mplsTunnelHopListIndexNext`--Next valid index to use as an index in the `mplsTunnelHopTable`.
- **`mplsTunnelHopTable`** --Entries in this table exist only for tunnel configurations and correspond to the path options defined for the tunnel. Two types of path options exist: *explicit* and *dynamic* . This table shows all hops listed in the explicit path options, while showing only the destination hop for dynamic path options. The tunnel hop table is indexed by tunnel ID, path option, and hop number.

Following is a list and description of each table entry.

- ◦ `mplsTunnelHopListIndex`--Primary index into the table.
- ◦ `mplsTunnelHopIndex`--Secondary index into the table.
- ◦ `mplsTunnelHopAddrType`--Indicates if the address of this hop is the type IPv4 or IPv6.
- ◦ `mplsTunnelHopIpv4Addr`--The IPv4 address of this hop.
- ◦ `mplsTunnelHopIpv4PrefixLen`--The prefix length of the IPv4 address.
- ◦ `mplsTunnelHopIpv6Addr`--The IPv6 address of this hop.
- ◦ `mplsTunnelHopIpv6PrefixLen`--The prefix length of the IPv6 address.
- ◦ `mplsTunnelHopAsNumber`--This object will contain 0 or the autonomous system number of the hop, depending on the value of `mplsTunnelHopAddrType`.
- ◦ `mplsTunnelHopLspId`--This object will contain 0 or the LSPID of the tunnel, depending on the value of `mplsTunnelHopAddrType`.
- ◦ `mplsTunnelHopType`--Denotes whether this tunnel hop is routed in a strict or loose fashion.
- ◦ `mplsTunnelHopRowStatus`--This object is used in conjunction with the configuring of a new row in the table.
- ◦ `mplsTunnelHopStorageType`--The storage type of this MIB object.
- `mplsTunnelResourceIndexNext`--This object contains the next appropriate value to be used for `mplsTunnelResourceIndex` when creating entries in the `mplsTunnelResourceTable`
- **`mplsTunnelResourceTable`** --Entries in this table correspond to the "Tspec" information displayed when you execute the `show mpls traffic-eng tunnels` command. These entries exist only for LSPs.

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

Following is a list and description of each table entry.

- - `mplsTunnelResourceIndex`--The primary index into this table.
 - `mplsTunnelResourceMaxRate`--The maximum rate, in bits per second, supported by this tunnel.
 - `mplsTunnelResourceMeanRate`--The mean rate, in bits per second, supported by this tunnel.
 - `mplsTunnelResourceMaxBurstSize`--The maximum burst size, in bytes, allowed by this tunnel.
 - `mplsTunnelResourceRowStatus`--This object is used in conjunction with the configuration of a new row in the table.
 - `mplsTunnelResourceStorageType`--The storage type of this MIB object.
- **mplsTunnelARHopTable** --Entries in this table correspond to the actual route taken by the tunnel, and whose route was successfully signaled by the network. The hops present in this table correspond to those present in the record route object (RRO) in Resource Reservation Protocol (RSVP). You can also display the information in this table by executing the `show mpls traffic-eng tunnels` command.

The actual route hop table is indexed by address and hop number. Following the `mplsTunnelARHopTableIndex` pointer from the tunnel table is the best way to retrieve information from this table.

Following is a list and description of each table entry:

- - `mplsTunnelARHopListIndex`--The primary index into this table.
 - `mplsTunnelARHopIndex`--The secondary index into this table.
 - `mplsTunnelARHopIpv4Addr`--The IPv4 address of this hop.
 - `mplsTunnelARHopIpv4PrefixLen`--The prefix length of the IPv4 address.
 - `mplsTunnelARHopIpv6Addr`--The IPv6 address of this hop.
 - `mplsTunnelARHopIpv6PrefixLen`--The prefix length of the IPv6 address.
 - `mplsTunnelARHopAsNumber`--This object will contain 0 or the AS number of the hop, depending on the value of `mplsTunnelARHopAddrType`.
 - `mplsTunnelARHopAddrType`--The type of address for this MIB entry, either IPv4 or IPv6.
 - `mplsTunnelARHopType`--Denotes whether this tunnel hop is routed in a strict or loose manner.
- **mplsTunnelCHopTable** --Entries in this table correspond to the explicit route object (ERO) in RSVP, which is used to signal the LSP. The list of hops in this table will contain those hops that are computed by the constraint-based shortest path first (SPF) algorithm. In those cases where “loose” hops are specified for the tunnel, this table will contain the hops that are “filled-in” between the loose hops to complete the path. If you specify a complete explicit path, the computed hop table matches your specified path.

The computed hop table is indexed by address and hop number. Following the `mplsTunnelCHopTableIndex` pointer from the tunnel table is the best way to retrieve information from this table.

Following is a list and description of each table entry:

- - `mplsTunnelCHopListIndex`--The primary index into this table.
 - `mplsTunnelCHopIndex`--The secondary index into this table.
 - `mplsTunnelCHopAddrType`--Indicates if the address of this hop is the type IPv4 or IPv6.
 - `mplsTunnelCHopIpv4Addr`--The IPv4 address of this hop.
 - `mplsTunnelCHopIpv4PrefixLen`--The prefix length of the IPv4 address.
 - `mplsTunnelCHopIpv6Addr`--The IPv6 address of this hop.

- `mplsTunnelCHopIpv6PrefixLen`--The prefix length of the IPv6 address.
- `mplsTunnelCHopAsNumber`--This object will contain 0 or the autonomous system number of the hop, depending on the value of `mplsTunnelHopAddrType`.
- `mplsTunnelCHopType`--Denotes whether this tunnel hop is routed in a strict or loose way.
- **`mplsTunnelPerfTable`** --The tunnel performance table, which augments the **`mplsTunnelTable`**, provides packet and byte counters for each tunnel. This table contains the following packet and byte counters:
 - `mplsTunnelPerfPackets`--This packet counter works only for tunnel heads.
 - `mplsTunnelPerfHCPackets`--This packet counter works only for tunnel heads.
 - `mplsTunnelPerfErrors`--This packet counter works only for tunnel heads.
 - `mplsTunnelPerfBytes`--This byte counter works for tunnel heads and tunnel midpoints, but not for tunnel tails.
 - `mplsTunnelPerfHCBytes`--This byte counter works for tunnel heads and tunnel midpoints, but not for tunnel tails.
- `mplsTunnelTrapEnable`--The object type *mplsTunnelTrapEnable* is enhanced to be writable. Accordingly, if this object type is set to “TRUE,” the following notifications are enabled, thus giving you the ability to monitor changes in the operational status of MPLS TE tunnels:
 - `mplsTunnelUp`
 - `mplsTunnelDown`
 - `mplsTunnelRerouted`

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

CLI Access to MPLS Traffic Engineering MIB Information

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

Figure 11 Commands for Retrieving MPLS TE MIB Information

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

- [Retrieving Information from the MPLS Traffic Engineering MIB, page 87](#)

Retrieving Information from the MPLS Traffic Engineering MIB

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

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

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

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

How to Configure the MPLS Traffic Engineering MIB

- [Enabling the SNMP Agent to Help Manage Various MPLS TE Tunnel Characteristics of Tunnels on the Local Router](#), page 88
- [Verifying the Status of the SNMP Agent](#), page 89

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>telnet host</code> Example: <pre>Router> telnet 192.172.172.172</pre>	Telnet to the router identified by the specified IP address (represented as xxx.xxx.xxx.xxx).
Step 2	<code>enable</code> Example: <pre>Router# enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 3	<code>show running-config</code> Example: <pre>Router# show running-config</pre>	Displays the running configuration to determine if an SNMP agent is already running. <ul style="list-style-type: none"> • If no SNMP information is displayed, go to Step 4 . If any SNMP information is displayed, you can modify the information or change it as needed.

Command or Action	Purpose
Step 4 <code>configure terminal</code> Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 5 <code>snmp-server community string [view view-name] [ro rw] [ipv6 nacl] [access-list-number]</code> Example: <pre>Router(config)# snmp-server community comaccess ro 4</pre>	Enables the read-only (RO) community string.
Step 6 <code>snmp-server enable traps [identification-type] [notification-option]</code> Example: <pre>Router(config)# snmp-server enable traps</pre>	Enables an LSR to send SNMP notifications or informs to an SNMP host. Note This command is optional. After SNMP is enabled, all MIBs (not just the TE MIB) are available for the user to query.
Step 7 <code>exit</code> Example: <pre>Router(config)# exit</pre>	Exits global configuration mode and returns to privileged EXEC mode.
Step 8 <code>write memory</code> Example: <pre>Router# write memory</pre>	Writes the modified configuration to NVRAM, permanently saving the settings.

Verifying the Status of the SNMP Agent

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

SUMMARY STEPS

1. `telnet host`
2. `enable`
3. `show running-config`

DETAILED STEPS

Command or Action	Purpose
Step 1 <code>telnet host</code> Example: Router# <code>telnet 192.172.172.172</code>	Telnet to the target device identified by the specified IP address (represented as <code>xxx.xxx.xxx.xxx</code>).
Step 2 <code>enable</code> Example: Router# <code>enable</code>	Enables SNMP on the target device.
Step 3 <code>show running-config</code> Example: Router# <code>show running-config</code>	Displays the running configuration on the target device and is used to examine the output for displayed SNMP information.

- [Examples, page 90](#)

Examples

The following example displays the running configuration on the target device and its SNMP information.

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

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

Configuration Examples for the MPLS Traffic Engineering MIB

- [Example Enabling the SNMP Agent to Help Manage MPLS TE Characteristics of Tunnels on the Local Router, page 91](#)

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

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

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

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

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

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

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

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Information about MPLS TE and enhancements	MPLS Traffic Engineering and Enhancements
MPLS TE commands	<i>Multiprotocol Label Switching Command Reference</i>
SNMP commands	<i>Network Management Command Reference</i>
SNMP configuration	“Configuring SNMP Support” in the <i>Network Management Configuration Guide</i>

Standards

Standard	Title
draft-ietf-mpls-te-mib-05	<i>MPLS Traffic Engineering Management Information Base Using SMIv2</i>

MIBs

MIB	MIBs Link
<ul style="list-style-type: none"> MPLS TE MIB Interfaces MIB 	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2026	<i>The Internet Standards Process</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for the MPLS Traffic Engineering 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 18 Feature Information for the MPLS Traffic Engineering MIB

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

Glossary

affinity bits --A Multiprotocol Label Switching (MPLS) traffic engineering (TE) tunnel's requirements on the attributes of the links it will cross. The tunnel's affinity bits and affinity mask must match with the attributes of the various links carrying the tunnel.

call admission precedence --An Multiprotocol Label Switching (MPLS) traffic engineering tunnel with a higher priority will, if necessary, preempt an MPLS traffic engineering tunnel with a lower priority. An

expected use is that tunnels that are more difficult to route will have a higher priority, and can preempt tunnels that are less difficult to route, on the assumption that those lower priority tunnels can find another path.

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

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

headend --The label switch router (LSR) at which the tunnel originates. The tunnel's "head" or tunnel interface will reside at this LSR as well.

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

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

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

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

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

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

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

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.

notification --A message sent by a Simple Network Management Protocol (SNMP) agent to a network management station, console, or terminal to indicate that a significant event within Cisco software has occurred (see traps).

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

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

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

tailend --The downstream, receive end of a tunnel.

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

trap --A message sent by a Simple Network Management Protocol (SNMP) agent to a network management station, console, or terminal to indicate that a significant event within Cisco software has

occurred. Traps (notifications) are less reliable than inform requests, because the receiver of the trap does not send an acknowledgment of receipt; furthermore, the sender of the trap cannot determine if the trap was received (see notification).

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

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

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

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

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

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



Monitoring MPLS VPNs with MIBs

This module explains how to use the PPVPN-MPLS-VPN management information base (MIB) to monitor and manage Multiprotocol Label Switching (MPLS) Virtual Private Networks. The following MIBs are supported:

- *MPLS/BGP Virtual Private Network Management Information Base Using SMIPv2 (draft-ietf-ppvpn-mpls-vpn-mib-03.txt)*
- CISCO-IETF-PPVPN-MPLS-VPN-MIB, a proprietary MIB that describes the cMplsNumVrfRouteMaxThreshCleared notification.
- [Finding Feature Information, page 97](#)
- [Prerequisites for PPVPN MPLS VPN MIB, page 97](#)
- [Restrictions for PPVPN MPLS VPN MIB, page 98](#)
- [Information About PPVPN MPLS VPN MIB, page 98](#)
- [How to Configure PPVPN MPLS VPN MIB, page 115](#)
- [Configuration Examples for PPVPN MPLS VPN MIB, page 121](#)
- [Additional References, page 122](#)
- [Feature Information for PPVPN MPLS VPN MIB, page 123](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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.

Prerequisites for PPVPN MPLS VPN MIB

The PPVPN-MPLS-VPN MIB agent requires the following:

- SNMP is installed and enabled on the label switching routers.
- MPLS is enabled on the label switching routers.
- Multiprotocol Border Gateway Protocol (BGP) is enabled on the label switching routers.
- Cisco Express Forwarding is enabled on the label switching routers.

Restrictions for PPVPN MPLS VPN MIB

- Configuration of the MIB using the SNMP SET command is not supported, except for trap-related objects, such as `mplsVpnNotificationEnable` and `mplsVpnVrfSecIllegalLabelRcvThresh`.
- The `mplsVpnVrfBgpNbrPrefixTable` is not supported.

Information About PPVPN MPLS VPN MIB

- [MPLS VPN Overview](#), page 98
- [PPVPN MPLS VPN MIB Overview](#), page 98
- [PPVPN MPLS VPN MIB and the IETF](#), page 99
- [Capabilities Supported by PPVPN-MPLS-VPN MIB](#), page 99
- [Functional Structure of the PPVPN-MPLS-VPN MIB](#), page 99
- [Supported Objects in PPVPN-MPLS-VPN MIB](#), page 100
- [MIB Objects Not Supported](#), page 114

MPLS VPN Overview

The MPLS VPN technology allows service providers to offer intranet and extranet VPN services that directly connect their customers' remote offices to a public network with the same security and service levels that a private network offers. Each VPN is associated with one or more VPN routing/forwarding instances (VRFs). A VRF is created for each VPN defined on a router and contains most of the information needed to manage and monitor MPLS VPNs: an IP routing table, a derived Cisco Express Forwarding (CEF) table, a set of interfaces that use this forwarding table, and a set of rules and routing protocol parameters that control the information that is included in the routing table.

PPVPN MPLS VPN MIB Overview

The Provider-Provisioned VPN (PPVPN)-MPLS-VPN MIB provides access to VRF information, as well as interfaces included in the VRF, and other configuration and monitoring information.

The PPVPN-MPLS-VPN MIB provides the following benefits:

- A standards-based SNMP interface for retrieving information about critical MPLS VPN events.
- VRF information to assist in the management and monitoring of MPLS VPNs.
- Information, in conjunction with the Interfaces MIB, about interfaces assigned to VRFs.
- Performance statistics for all VRFs on a router.
- The generation and queuing of notifications that call attention to major changes in the operational status of MPLS VPN enabled interfaces; the forwarding of notification messages to a designated network management system (NMS) for evaluation and action by network administrators.
- Advanced warning when VPN routing tables are approaching or exceed their capacity.
- Warnings about the reception of illegal labels on a VRF-enabled interface. Such receptions may indicate misconfiguration or an attempt to violate security.

This document also describes the CISCO-IETF-PPVPN-MPLS-VPN-MIB, which contains the `cMplsNumVrfRouteMaxThreshCleared` notification.

PPVPN MPLS VPN MIB and the IETF

SNMP agent code operating with the PPVPN-MPLS-VPN MIB enables a standardized, SNMP-based approach to managing MPLS VPNs in Cisco IOS software.

The PPVPN-MPLS-VPN MIB is based on the IETF draft MIB specification *draft-ietf-ppvpn-mpls-vpn-mib-03.txt*, which includes objects describing features that support MPLS VPN events. This IETF draft MIB, which undergoes revisions from time to time, is being evolved toward becoming a standard. Accordingly, the Cisco implementation of the PPVPN-MPLS-VPN MIB is expected to track the evolution of the IETF draft MIB, and may change accordingly.

Some slight differences between the IETF draft MIB and the actual implementation of MPLS VPNs within Cisco IOS software require some minor translations between the PPVPN-MPLS-VPN MIB and the internal data structures of Cisco IOS. These translations are accomplished by means of the SNMP agent code. Also, while running as a low priority process, the SNMP agent provides a management interface to Cisco IOS. SNMP adds little overhead on the normal functions of the device.

The SNMP objects defined in the PPVPN-MPLS-VPN MIB can be viewed by any standard SNMP utility. The network administrator can retrieve information in the PPVPN-MPLS-VPN MIB using standard SNMP **get** and **getnext** operations for SNMP v1, v2, and v3.

All PPVPN-MPLS-VPN MIB objects are based on the IETF draft MIB; thus, no Cisco specific SNMP application is required to support the functions and operations pertaining to the PPVPN-MPLS-VPN MIB features.

Capabilities Supported by PPVPN-MPLS-VPN MIB

The following functionality is supported by the PPVPN-MPLS-VPN MIB. The PPVPN-MPLS-VPN MIB provides you with the ability to do the following:

- Gather routing and forwarding information for MPLS VPNs on a router.
- Expose information in the VRF routing table.
- Gather information on BGP configuration related to VPNs and VRF interfaces and statistics.
- Emit notification messages that signal changes when critical MPLS VPN events occur.
- Enable, disable, and configure notification messages for MPLS VPN events by using extensions to existing SNMP CLI commands.
- Specify the IP address of a network management system (NMS) in the operating environment to which notification messages are sent.
- Write notification configurations into nonvolatile memory.

Functional Structure of the PPVPN-MPLS-VPN MIB

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

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

- Platform-independent layer--This layer is generated primarily by the MIB development Cisco tool set and incorporates platform- and implementation-independent functions. The Cisco MIB development tool set creates a standard set of files associated with a MIB.
- Application interface layer--The functions, names, and template code for MIB objects in this layer are also generated by the MIB development Cisco tool set.

- Application-specific layer--This layer provides an interface between the application interface layer and the API and data structures layer below and performs tasks needed to retrieve required information from Cisco software, such as searching through data structures.
- API and data structures layer--This layer contains the data structures or APIs within Cisco software that are retrieved or called in order to set or retrieve SNMP management information.

Supported Objects in PPVPN-MPLS-VPN MIB

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

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

The PPVPN-MPLS-VPN MIB tables and objects are described briefly in the following sections:

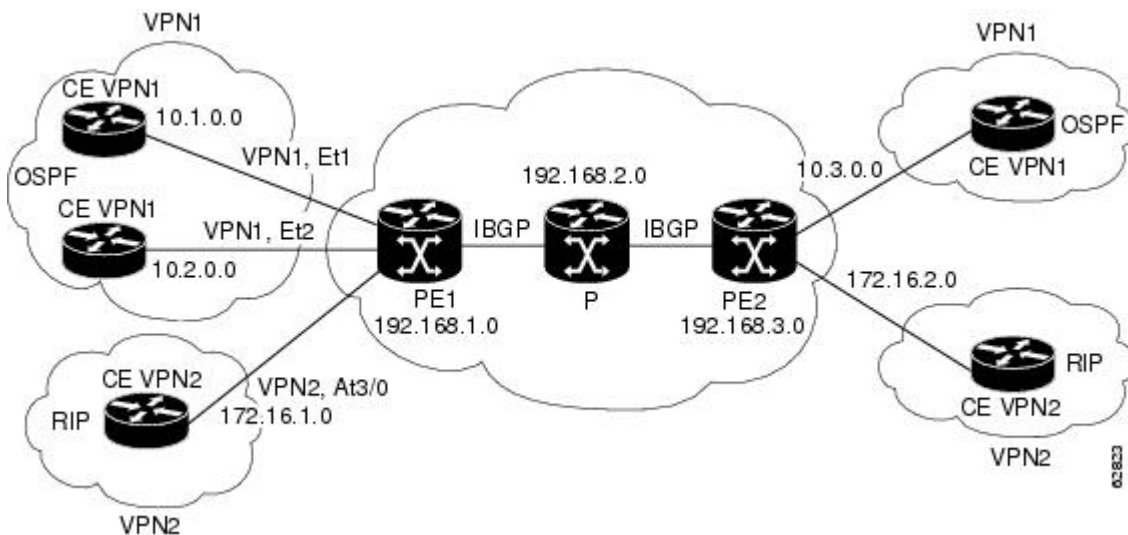
Objects that are not supported are listed in the MIB Objects Not Supported section.

The figure below shows a simple MPLS VPN configuration. This configuration includes two customer MPLS VPNs, labeled VPN1 and VPN2, and a simple provider network that consists of two provider edge (PE) routers, labeled PE1 and PE2, and a provider core router labeled P. The figure shows the following sample configuration:

- VRF names--VPN1 and VPN2
- Interfaces associated with VRFs--Et1, Et2, and At3/0
- Routing protocols--OSPF, RIP, and IBGP
- Routes associated with VPN1--10.1.0.0, 10.2.0.0, and 10.3.0.0
- Routes associated with VPN2--172.16.1.0 and 172.16.2.0
- Routes associated with the provider network--192.168.1.0, 192.168.2.0, and 192.168.3.0

This configuration is used in this document to explain MPLS VPN events that are monitored and managed by the PPVPN-MPLS-VPN MIB.

Figure 12 Sample MPLS VPN Configuration



- [Scalar Objects, page 101](#)
- [MIB Tables, page 101](#)
- [Notifications, page 112](#)

Scalar Objects

The table below shows the supported PPVPN-MPLS-VPN MIB scalar objects.

Table 19 *PPVPN-MPLS-VPN MIB Scalar Objects*

MIB Object	Function
mplsVpnConfiguredVrfs	The number of VRFs configured on the router, including VRFs recently deleted.
mplsVpnActiveVrfs	The number of VRFs that are active on the router. An active VRF is assigned to at least one interface that is in the operationally up state.
mplsVpnConnectedInterfaces	The total number of interfaces assigned to any VRF.
mplsVpnNotificationEnable	<p>A value that indicates whether all the PPVPN-MPLS-VPN MIB notifications are enabled.</p> <ul style="list-style-type: none"> • Setting this object to true enables all notifications defined in the PPVPN-MPLS-VPN MIB. • Setting it to false disables all notifications defined in the MIB. <p>This is one of the few objects that is writable.</p>
mplsVpnVrfConfMaxPossibleRoutes	A number that indicates the amount of routes that this router is capable of storing. This value cannot be determined because it is based on the amount of available memory in the system. Therefore, this object is set to zero (0).

MIB Tables

The PPVPN-MPLS-VPN MIB implementation supports the following tables described in this section:

- [mplsVpnVrfTable, page 101](#)
- [mplsVpnInterfaceConfTable, page 104](#)
- [mplsVpnVrfRouteTargetTable, page 105](#)
- [mplsVpnVrfBgpNbrAddrTable, page 107](#)
- [mplsVpnVrfSecTable, page 108](#)
- [mplsVpnVrfPerfTable, page 108](#)
- [mplsVpnVrfRouteTable, page 109](#)

mplsVpnVrfTable

Entries in the VRF configuration table (mplsVpnVrfTable) represent the VRFs that are defined on the router. This includes recently deleted VRFs. The information in this table is also displayed with the **show ip vrf** command.

Each VRF is referenced by its VRF name (mplsVpnVrfName).

The table below lists the MIB objects and their functions for this table.

Table 20 PPVPN-MPLS-VPN MIB Objects for the mplsVpnVrfTable

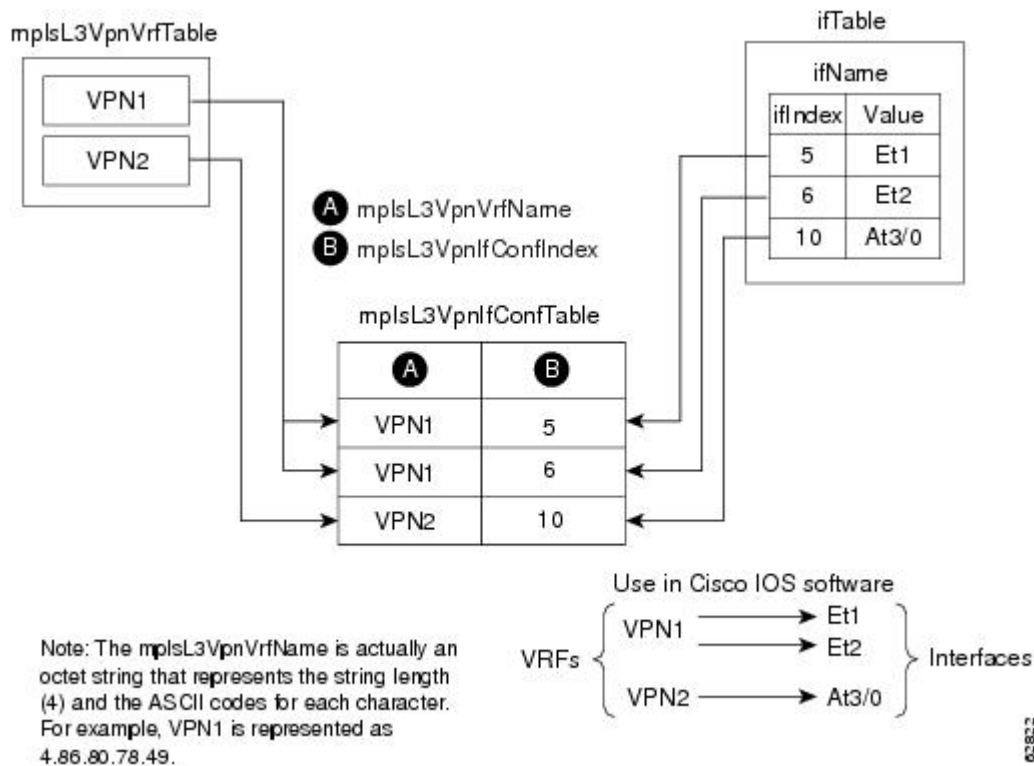
MIB Object	Function
mplsVpnVrfName	The name associated with this VRF. When this object is used as an index to a table, the first octet is the string length, and subsequent octets are the ASCII codes of each character. For example, "vpn1" is represented as 4.118.112.110.49.
mplsVpnVrfDescription	The description of the VRF. This is specified with the following configuration command: <pre>Router(config)# ip vrf vrf-name Router(config-vrf)# description vrf-description</pre>
mplsVpnVrfRouteDistinguisher	The route distinguisher for this VRF. This is specified with the following configuration command: <pre>Router(config)# ip vrf vrf-name Router(config-vrf)# rd route-distinguisher</pre>
mplsVpnVrfCreationTime	The value of the sysUpTime when this VRF entry was created.
mplsVpnVrfOperStatus	The operational status of this VRF. A VRF is up (1) when at least one interface associated with the VRF is up. A VRF is down (2) when: <ul style="list-style-type: none"> • No interfaces exist whose ifOperStatus = up (1). • No interfaces are associated with this VRF.
mplsVpnVrfActiveInterfaces	The number of interfaces assigned to this VRF which are operationally up.
mplsVpnVrfAssociatedInterfaces	The number of interfaces assigned to this VRF, independent of the operational status.

MIB Object	Function
mplsVpnVrfConfMidRouteThreshold	<p>The middle route threshold. If the amount of routes in the VRF crosses this threshold, an <code>mplsNumVrfRouteMidThreshExceeded</code> notification is sent (if notifications are enabled and configured). You can set this value in configuration mode as a percentage of the maximum with the maximum routes limit <code>{warn-threshold warn-only}</code> command, as follows:</p> <pre>Router(config)# ip vrf vpn1 Router(config-vrf)# maximum routes 1000 50</pre> <p>The middle or warn threshold is set for VRF <code>vpn1</code> as 50% of the maximum route threshold.</p> <p>The following command sets a middle threshold of 1000 routes. An <code>mplsNumVrfRouteMidThreshExceeded</code> notification is sent when this threshold is exceeded. However, additional routes are still allowed because a maximum route threshold is not set with this command.</p> <pre>Router(config-vrf)# maximum routes 1000 warn-only</pre>
mplsVpnVrfConfHighRouteThreshold	<p>The maximum route threshold. If the amount of routes in the VRF crosses this threshold, an <code>mplsNumVrfRouteMaxThreshExceeded</code> notification is sent (if notifications are enabled and configured). You can set this value in configuration mode with the maximum routes limit <code>{warn-threshold warn-only}</code> command as follows:</p> <pre>Router(config)# ip vrf vpn2 Router(config-vrf)# maximum routes 1000 75</pre> <p>The maximum route threshold is set for 1000 routes for VRF <code>vpn2</code> with a middle or warn threshold of 75% of this threshold.</p>
mplsVpnVrfConfMaxRoutes	<p>This value is the same as the <code>mplsVpnVrfConfHighRouteThreshold</code>.</p>
mplsVpnVrfConfLastChanged	<p>The value of <code>sysUpTime</code> when the configuration of the VRF changes or interfaces are assigned or unassigned from the VRF.</p> <p>Note This object is updated only when values in this table change.</p>
mplsVpnVrfConfRowStatus	<p>Read-only implementation. This object normally reads “active (1),” but may read “notInService (2),” if a VRF was recently deleted.</p>
mplsVpnVrfConfStorageType	<p>Read-only implementation. This object always reads “volatile (2).”</p>

mplsVpnInterfaceConfTable

In Cisco software, a VRF is associated with one MPLS VPN. Zero or more interfaces can be associated with a VRF. A VRF uses an interface that is defined in the ifTable of the Interfaces Group of MIB II (IFMIB). The IFMIB defines objects for managing interfaces. The ifTable of this MIB contains information on each interface in the network. The mplsVpnInterfaceConfTable associates a VRF from the mplsVpnVrfTable with a forwarding interface from the ifTable. The figure below shows the relationship between VRFs and interfaces defined in the ifTable and the mplsVpnInterfaceConfTable.

Figure 13 VRFs, the Interfaces MIB, and the mplsVpnInterfaceConfTable



Entries in the VPN interface configuration table (mplsVpnInterfaceConfTable) represent the interfaces that are assigned to each VRF. The information available in this table is also displayed with the **show ip vrf** command.

The mplsVpnInterfaceConfTable shows how interfaces are assigned to VRFs. A label switch router (LSR) creates an entry in this table for every interface capable of supporting MPLS VPNs.

The mplsVpnInterfaceConfTable is indexed by the following:

- mplsVpnVrfName--The VRF name
- mplsVpnInterfaceConfIndex--An identifier that is the same as the ifIndex from the Interface MIB of the interface assigned to the VRF

The table below lists the MIB objects and their functions for this table.

Table 21 *PPVPN-MPLS-VPN MIB Objects for the mplsVpnInterfaceConfTable*

MIB Object	Function
mplsVpnInterfaceConfIndex	Provides the interface MIB ifIndex of this interface that is assigned to a VRF.
mplsVpnInterfaceLabelEdgeType	Indicates whether the interface is a provider edge interface (1) or a customer edge interface (2). This value is always providerEdge (1) because in Cisco IOS, customerEdge interfaces are not assigned to VRFs and do not appear in this table.
mplsVpnInterfaceVpnClassification	Specifies what type of VPN this interface is providing: carrier supporting carrier (CsC) (1), enterprise (2), or InterProvider (3). This value is set to enterprise (2) if MPLS is not enabled and to carrier supporting carrier (1) if MPLS is enabled on this interface.
mplsVpnInterfaceVpnRouteDistProtocol	Indicates the route distribution protocols that are being used to redistribute routes with BGP on this interface: BGP (2), OSPF (3), or RIP (4). In Cisco software, router processes are defined and redistributed on a per-VRF basis, not per-interface. Therefore, all interfaces assigned to the same VRF have the same value for this object.
mplsVpnInterfaceConfStorageType	Read-only implementation. This object always reads “volatile (2).”
mplsVpnInterfaceConfRowStatus	Read-only implementation. This object normally reads “active (1),” but may read “notInService (2),” if a VRF was recently deleted.

mplsVpnVrfRouteTargetTable

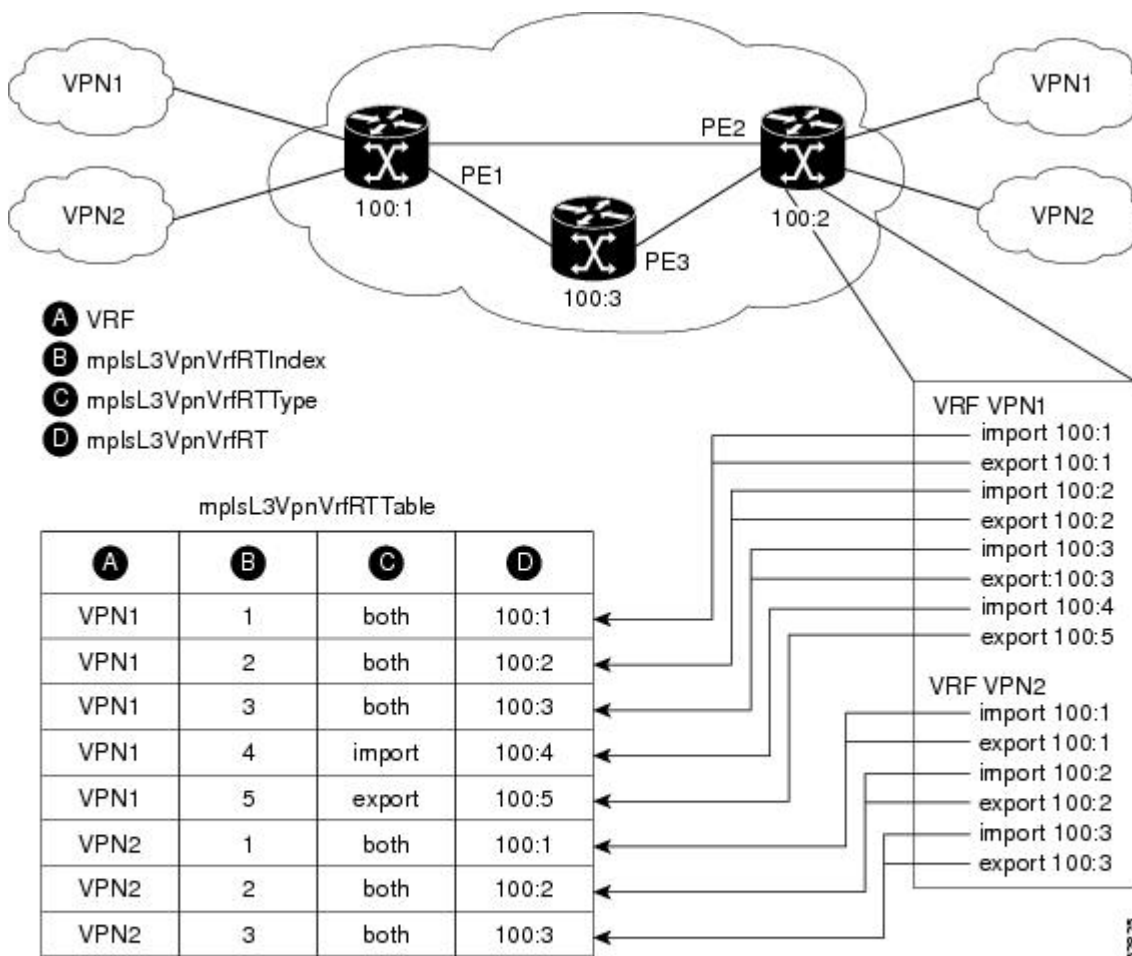
The route target table (mplsVpnVrfRouteTargetTable) describes the route target communities that are defined for a particular VRF. An LSR creates an entry in this table for each target configured for a VRF supporting an MPLS VPN instance.

The distribution of VPN routing information is controlled through the use of VPN route target communities, implemented by Border Gateway Protocol (BGP) extended communities. Distribution of VPN routing information works as follows:

- When a VPN route learned from a CE router is injected into BGP, a list of VPN route target extended community attributes are associated with it. Typically the list of route target community values is set from an export list of route targets associated with the VRF from which the route was learned.
- An import list of route target extended communities is associated with each VRF. The import list defines route target extended community attributes a route must have for the route to be imported into the VRF. For example, if the import list for a particular VRF includes route target communities A, B, and C, then any VPN route that carries any of those route target extended communities--A, B, or C--is imported into the VRF.

The figure below shows a sample configuration and its relationship to an mplsVpnVrfRouteTargetTable. A route target table exists on each PE router. Routers with route distinguishers (RDs) 100:1, 100:2, and 100:3 are shown in the sample configuration. Routers with RDs 100:4 and 100:5 are not shown in the figure, but are included in the route targets for PE2 and in the mplsVpnVrfRouteTargetTable.

Figure 14 Sample Configuration and the mplsVpnVrfRouteTargetTable



Note: The mplsL3VpnVrfName is actually an octet string that represents the string length (4) and the ASCII codes for each character. For example, VPN1 is represented as 4.86.80.78.49.

The mplsVpnVrfRouteTargetTable shows the import and export route targets for each VRF. The table is indexed by the following:

- mplsVpnVrfName--The VRF name
- mplsVpnVrfRouteTargetIndex--The route target entry identifier
- mplsVpnVrfRouteTargetType--A value specifying whether the entry is an import route target, export route target, or is defined as both

The table below lists the MIB objects and their functions for this table.

Table 22 *PPVPN-MPLS-VPN MIB Objects for the mplsVpnVrfRouteTargetTable*

MIB Object	Function
mplsVpnVrfRouteTargetIndex	A value that defines each route target's position in the table.
mplsVpnVrfRouteTargetType	Determines which type of route target the entry represents: import (1), export (2), or both (3).
mplsVpnVrfRouteTarget	Determines the route distinguisher for this target.
mplsVpnVrfRouteTargetDescr	Description of the route target. This object is not supported. Therefore, the object is the same as mplsVpnVrfRouteTarget.
mplsVpnVrfRouteTargetRowStatus	Read-only implementation. This object normally reads "active (1)," but may read "notInService (2)," if a VRF was recently deleted.

mplsVpnVrfBgpNbrAddrTable

The BGP neighbor address table (mplsVpnVrfBgpNbrAddrTable) represents the MPLS eBGP neighbors that are defined for a particular VRF. An LSR creates an entry for every BGP neighbor that is defined in the VRF's address-family.

The mplsVpnVrfBgpNbrAddrTable is indexed by the following:

- mplsVpnVrfName--The VRF name
- mplsVpnInterfaceConfIndex--An identifier that is the same as the ifIndex from the Interface MIB of the interface assigned to the VRF
- mplsVpnVrfBgpNbrIndex--The IP address of the neighbor

The table below lists the MIB objects and their functions for this table.

Table 23 *PPVPN-MPLS-VPN MIB Objects for the mplsVpnVrfBgpNbrAddrTable*

MIB Object	Function
mplsVpnVrfBgpNbrIndex	The IPv4 address of the eBGP neighbor.
mplsVpnVrfBgpNbrRole	The role of this eBGP neighbor: customer edge (1) or provider edge (2). If the object mplsVpnInterfaceVpnClassification is carrier supporting carrier (CSC), then this value is provider edge (2), otherwise, this value is customer edge (1).
mplsVpnVrfBgpNbrType	Address type of this eBGP neighbor. The MIB only supports IPv4 (1). Therefore, this object returns "ipv4 (1)."
mplsVpnVrfBgpNbrAddr	IP address of the eBGP neighbor.
mplsVpnVrfBgpNbrRowStatus	Read-only implementation. This object normally reads "active (1)," but may read "notInService (2)" if a VRF was recently deleted.

MIB Object	Function
mplsVpnVrfBgpNbrStorageType	Read-only implementation. This object always reads “volatile (2).”

mplsVpnVrfSecTable

The VRF security table (mplsVpnVrfSecTable) provides information about security for each VRF. An LSR creates an entry in this table for every VRF capable of supporting MPLS VPN.

The mplsVpnVrfSecTable *augments* the mplsVpnVrfTable and has the same indexing.

The table below lists the MIB objects and their functions for this table.

Table 24 *PPVPN-MPLS-VPN MIB Objects for the mplsVpnVrfSecTable*

MIB Object	Function
mplsVpnVrfSecIllegalLabelViolations	<p>The number of illegally received labels on a VRF interface. Only illegal labels are counted by this object, therefore the object only applies to a VRF interface that is MPLS enabled (carrier supporting carrier [CsC] situation).</p> <p>This counter is incremented whenever a label is received that is above or below the valid label range, not in the global label forwarding table, or is received on the wrong VRF (that is, table IDs for the receiving interface and appropriate VRF label forwarding table do not match).</p>
mplsVpnVrfSecIllegalLabelRcvThresh	<p>Notification threshold for illegal labels received on this VRF. When the amount of illegal labels received on this interface crosses this threshold, an mplsNumVrfSecIllegalLabelThreshExceeded notification is sent (if the notification is enabled and configured).</p> <p>This object is one of the few in this MIB agent that supports the SNMP SET operation, which allows you to change this value.</p>

mplsVpnVrfPerfTable

The VRF performance table (mplsVpnVrfPerfTable) provides statistical performance information for each VRF. An LSR creates an entry in this table for every VRF capable of supporting MPLS VPN.

The mplsVpnVrfPerfTable *augments* the mplsVpnVrfTable and has the same indexing.

The table below lists the MIB objects and their functions for this table.

Table 25 *PPVPN-MPLS-VPN MIB Objects for the mplsVpnVrfPerfTable*

MIB Objects	Functions
mplsVpnVrfPerfRoutesAdded	The number of routes added to this VRF over the course of its lifetime.

MIB Objects	Functions
mplsVpnVrfPerfRoutesDeleted	The number of routes removed from this VRF.
mplsVpnVrfPerfCurrNumRoutes	The number of routes currently defined within this VRF.

mplsVpnVrfRouteTable

The VRF routing table (mplsVpnVrfRouteTable) provides the IP routing table information for each VRF. The information available in this table can also be accessed with the **show ip route vrf vrf-name** command. For example, for PE1 in the figure above:

- With the **show ip route vrf vpn1** command, you would see results like the following:

```
Router# show ip route vrf vpn1
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
!
Gateway of last resort is not set
!
      10.0.0.0/32 is subnetted, 3 subnets
B       10.3.0.0 [200/0] via 192.168.2.1, 04:36:33
C       10.1.0.0/16 is directly connected, Ethernet1
C       10.2.0.0/16 [200/0] directly connected Ethernet2, 04:36:33
```

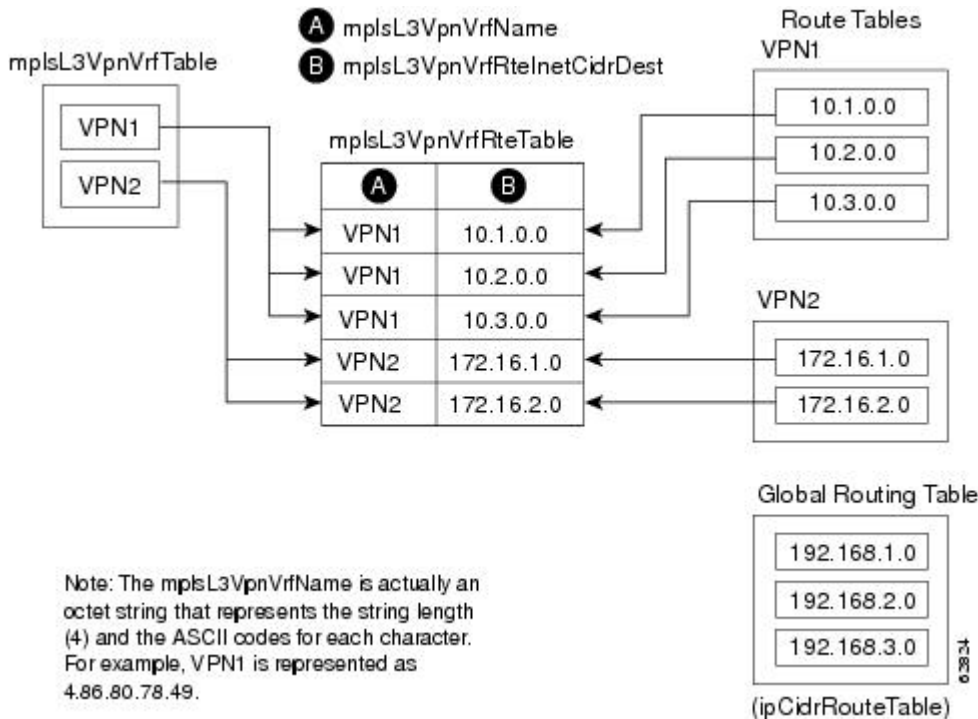
- With the **show ip route vrf vpn2** command, you would see results like the following:

```
Router# show ip route vrf vpn2
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
!
Gateway of last resort is not set
!
      172.16.0.0/32 is subnetted, 2 subnets
B       172.16.2.0 [200/0] via 192.168.2.1, 04:36:33
C       172.16.1.0 is directly connected, ATM 3/0
```

The figure below shows the relationship of the routing tables, the VRFs, and the mplsVpnVrfRouteTable. You can view information about the VPN1 and VPN2 route tables using the **show ip route vrf vrf-name**

command. The global route table is the same as ipCidrRouteTable in the IP-FORWARD-MIB. You can view information about the global route table with the **show ip route** command.

Figure 15 Route Table, VRFs, and the mplsVpnVrfRouteTable



An LSR creates an entry in this table for every route that is configured, either dynamically or statically, within the context of a specific VRF capable of supporting MPLS VPN.

The mplsVpnVrfRouteTable is indexed by the following:

- mplsVpnVrfName--The VRF name, which provides the VRF routing context
- mplsVpnVrfRouteDest--The IP destination address
- mplsVpnVrfRouteMask--The IP destination mask
- mplsVpnVrfRouteTos--The IP header ToS bits
- mplsVpnVrfRouteNextHop--The IP address of the next hop for each route entry



Note

The ToS bits are not supported and, therefore, are always 0.

The table below lists the MIB objects and their functions for the mplsVpnVrfRouteTable. This table represents VRF-specific routes. The global routing table is the ipCidrRouteTable in the IP-FORWARD-MIB.

Table 26 PPVPN-MPLS-VPN MIB Objects for the mplsVpnVrfRouteTable

MIB Object	Function
mplsVpnVrfRouteDest	The destination IP address defined for this route.

MIB Object	Function
mplsVpnVrfRouteDestAddrType	The address type of the IP destination address (mplsVpnVrfRouteDest). This MIB implementation only supports IPv4 (1). Therefore, this object has a value of “ipv4 (1).”
mplsVpnVrfRouteMask	The destination IP address mask defined for this route.
mplsVpnVrfRouteMaskAddrType	The address type of the destination IP address mask. This MIB implementation only supports IPv4 (1). Therefore, this object has a value of “ipv4 (1).”
mplsVpnVrfRouteTos	The ToS bits from the IP header for this route. Cisco software only supports ToS bits of zero. Therefore, the object is always 0.
mplsVpnVrfRouteNextHop	The next hop IP address defined for this route.
mplsVpnVrfRouteNextHopAddrType	The address type of the next hop IP address. This MIB implementation only supports IPv4 (1). Therefore, this object has a value of “ipv4 (1).”
mplsVpnVrfRouteIfIndex	The interface MIB ifIndex for the interface through which this route is forwarded. The object is 0 if no interface is defined for the route.
mplsVpnVrfRouteType	Defines if this route is a local or remotely defined route.
mplsVpnVrfRouteProto	The routing protocol that was responsible for adding this route to the VRF.
mplsVpnVrfRouteAge	The number of seconds since this route was last updated.
mplsVpnVrfRouteInfo	A pointer to more information from other MIBs. This object is not supported and always returns “nullOID (0.0).”
mplsVpnVrfRouteNextHopAS	The autonomous system number of the next hop for this route. This object is not supported and is always 0.
mplsVpnVrfRouteMetric1	The primary routing metric used for this route.
mplsVpnVrfRouteMetric2 mplsVpnVrfRouteMetric3 mplsVpnVrfRouteMetric4 mplsVpnVrfRouteMetric5	Alternate routing metrics used for this route. These objects are supported only for Cisco IGRP and Cisco EIGRP. These objects display the bandwidth metrics used for the route. Otherwise, these values are set to -1.
mplsVpnVrfRouteRowStatus	Read-only implementation. This object normally reads “active (1),” but may read “notInService (2),” if a VRF was recently deleted.
mplsVpnVrfRouteStorageType	Read-only implementation. This object always reads “volatile (2).”

Notifications

This section provides the following information about supported PPVPN-MPLS-VPN MIB notifications:

- [PPVPN-MPLS-VPN MIB Notification Events, page 112](#)
- [CISCO-IETF-PPVPN-MPLS-VPN MIB Notification Events, page 113](#)
- [Notification Specification, page 114](#)
- [Monitoring the PPVPN-MPLS-VPN MIB Notifications, page 114](#)

PPVPN-MPLS-VPN MIB Notification Events

The following notifications of the PPVPN-MPLS-VPN MIB are supported:

- **mplsVrflfUp** --Sent to an NMS when an interface comes up and is assigned a VPN routing/forwarding table instance (VRF).
- **mplsVrflfDown** --Generated and sent to the NMS when a VRF is removed from an interface or the interface transitions from an operationally “up” state to a “down” state.
- **mplsNumVrfRouteMidThreshExceeded** --Generated and sent when the middle (warning) threshold is crossed. You can configure this threshold in the CLI by using the following commands:

```
Router(config)# ip vrf
vrf-name
Router(config-vrf)# maximum routes
limit warn-threshold
(
% of max
)
```

The *warn-threshold* argument is a percentage of the maximum routes specified by the *limit* argument. You can also configure a middle threshold with the following command, in which the *limit* argument represents the warning threshold:

```
Router(config-vrf)# maximum routes
limit
warn-only
```

This notification is sent to the NMS only at the time the threshold is exceeded. (See the figure below for a comparison of the warning and maximum thresholds.) Whenever the number of routes falls below this threshold and exceeds the threshold again, a notification is sent to the NMS.

- **mplsNumVrfRouteMaxThreshExceeded** --Generated and sent when you attempt to create a route on a VRF that already contains the maximum number of routes as defined by the *limit* argument of the **maximum routes** commands:

```
Router(config)# ip vrf
vrf-name
Router(config-vrf)# maximum routes
limit
warn-threshold (% of max)
```

A trap notification is sent to the NMS when you attempt to exceed the maximum threshold. Another **mplsNumVrfRouteMaxThreshExceeded** notification is not sent until the number of routes falls below the maximum threshold and reaches the maximum threshold again. (See the figure below for an example of how this notification works and for a comparison of the maximum and warning thresholds.)

**Note**

The **maximum routes** command sets the number of routes for a VRF. You *cannot* exceed the number of routes in the VRF that you set with the **maximum routes limit warn-threshold** command. Prior to this implementation of the PPVPN-MPLS-VPN MIB, you were not notified when this threshold (or the warning threshold) was reached.

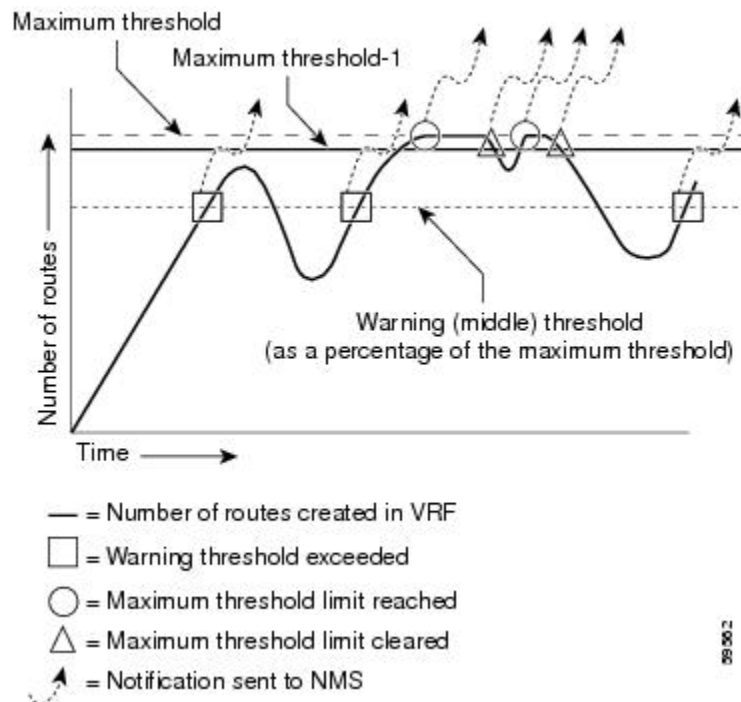
- **mplsNumVrfSecIllegalLabelThreshExceeded** --Generated and sent when the amount of illegal labels received on a VRF interface exceeds the threshold *mplsVpnVrfSecIllegalLabelRcvThresh* . This threshold is defined with a value of 0. Therefore, a notification is sent when the first illegal label is received on a VRF. Labels are considered illegal if they are outside of the valid label range, do not have a Label Forwarding Information Base (LFIB) entry, or the table ID of the message does not match the table ID for the label in the LFIB.

CISCO-IETF-PPVPN-MPLS-VPN MIB Notification Events

The following notification of the CISCO-IETF-PPVPN-MPLS-VPN MIB is supported in Cisco IOS Release 12.0(30)S:

- **cMplsNumVrfRouteMaxThreshCleared**--Generated and sent when the number of routes on a VRF attempts to exceed the maximum number of routes and then drops below the maximum number of routes. If you attempt to create a route on a VRF that already contains the maximum number of routes, the **mplsNumVrfRouteMaxThreshExceeded** notification is sent (if enabled). When you remove routes from the VRF so that the number of routes falls below the set limit, the **cMplsNumVrfRouteMaxThreshCleared** notification is sent. You can clear all routes from the VRF by using the **clear ip route vrf** command. (See the figure below to see when the **cMplsNumVrfRouteMaxThreshCleared** notification is sent.)

Figure 16 Comparison of Warning and Maximum Thresholds



Notification Specification

In an SNMPv1 notification, each VPN notification has a generic type identifier and an enterprise-specific type identifier for identifying the notification type.

- The generic type for all VPN notifications is “enterpriseSpecific” as this is not one of the generic notification types defined for SNMP.
- The enterprise-specific type is identified as follows:
 - 1 for *mplsVrfIfUp*
 - 2 for *mplsVrfIfDown*
 - 3 for *mplsNumVrfRouteMidThreshExceeded*
 - 4 for *mplsNumVrfRouteMaxThreshExceeded*
 - 5 for *mplsNumVrfSecIllegalLabelThreshExceeded*
 - 6 for *cMplsNumVrfRouteMaxThreshCleared*

In SNMPv2, the notification type is identified by an **SnmptTrapOID** varbind (variable binding consisting of an object identifier [OID] type and value) included within the notification message.

Each notification also contains two additional objects from the PPVPN-MPLS-VPN MIB. These objects provide additional information about the event, as follows:

- The VRF interface up/down notifications provide additional variables--*mplsVpnInterfaceConfIndex* and *mplsVpnVrfName*-- in the notification. These variables describe the SNMP interface index and the VRF name, respectively.
- The mid and max threshold notifications include the *mplsVpnVrfName* variable (VRF name) as well as the *mplsVpnVrfPerfCurrNumRoutes* variable that indicates the current number of routes within the VRF.
- The illegal label notification includes the *mplsVpnVrfName* variable (VRF name) and the *mplsVpnVrfSecIllegalLabelViolations* variable that maintains the current count of illegal labels on a VPN.

Monitoring the PPVPN-MPLS-VPN MIB Notifications

When PPVPN-MPLS-VPN MIB notifications are enabled (see the **snmp-server enable traps mpls vpn** command), notification messages relating to specific MPLS VPN events within Cisco software are generated and sent to a specified NMS in the network. Any utility that supports SNMPv1 or SNMPv2 notifications can receive notification messages.

To monitor PPVPN-MPLS-VPN MIB notification messages, log in to an NMS that supports a utility that displays SNMP notifications, and start the display utility.

MIB Objects Not Supported

The following objects from the *mplsVpnVrfBgpPathAttrTable* are not supported:

- *mplsVpnVrfBgpPathAttrPeer*
- *mplsVpnVrfBgpPathAttrIpAddrPrefixLen*
- *mplsVpnVrfBgpPathAttrIpAddrPrefix*
- *mplsVpnVrfBgpPathAttrOrigin*
- *mplsVpnVrfBgpPathAttrASPathSegment*
- *mplsVpnVrfBgpPathAttrNextHop*
- *mplsVpnVrfBgpPathAttrMultiExitDisc*

- `mplsVpnVrfBgpPathAttrLocalPref`
- `mplsVpnVrfBgpPathAttrAtomicAggregate`
- `mplsVpnVrfBgpPathAttrAggregatorAS`
- `mplsVpnVrfBgpPathAttrAggregatorAddr`
- `mplsVpnVrfBgpPathAttrCalcLocalPref`
- `mplsVpnVrfBgpPathAttrBest`
- `mplsVpnVrfBgpPathAttrUnknown`

How to Configure PPVPN MPLS VPN MIB

- [Configuring the SNMP Community, page 115](#)
- [Configuring the Router to Send SNMP Traps, page 116](#)
- [Configuring Threshold Values for MPLS VPN--SNMP Notifications, page 119](#)

Configuring the SNMP Community

An SNMP community string defines the relationship between the SNMP manager and the agent. The community string acts like a password to regulate access to the agent on the router.

Perform this task to configure an SNMP community.

SUMMARY STEPS

1. `enable`
2. `show running-config [options]`
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 [options]`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p><code>enable</code></p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p><code>show running-config [options]</code></p> <p>Example:</p> <pre>Router# show running-config</pre>	<p>Displays the running configuration to determine if an SNMP agent is already running.</p> <ul style="list-style-type: none"> • 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.

Command or Action	Purpose
Step 3 <code>configure terminal</code> Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 4 <code>snmp-server community string [view view-name] [ro rw] [acl=number]</code> Example: <pre>Router(config)# snmp-server community comaccess ro</pre>	Sets up the community access string to permit access to the SNMP protocol. <ul style="list-style-type: none"> The <i>string</i> argument acts like a password and permits access to the SNMP protocol. The view <i>view-name</i> keyword argument pair specifies the name of a previously defined view. The view defines the objects available to the community. The ro keyword specifies read-only access. Authorized management stations are only able to retrieve MIB objects. The rw keyword specifies read/write access. Authorized management stations are able to both retrieve and modify MIB objects. The <i>acl-number</i> 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 <code>do copy running-config startup-config</code> Example: <pre>Router(config)# do copy running- config startup-config</pre>	Saves the modified configuration to nonvolatile memory (NVRAM) as the startup configuration file. <ul style="list-style-type: none"> The do command allows you to perform EXEC level commands in configuration mode.
Step 6 <code>exit</code> Example: <pre>Router(config)# exit</pre>	Returns to privileged EXEC mode.
Step 7 <code>show running-config [options]</code> Example: <pre>Router# show-running config include snmp-server</pre>	(Optional) Displays the configuration information currently on the router, the configuration for a specific interface, or map-class information. <ul style="list-style-type: none"> 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 you define this string using the **snmp-server community** command before 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 vpn** [**illegal-label**] [**max-thresh-cleared**] [**max-threshold**] [**mid-threshold**] [**vrf-down**] [**vrf-up**]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

Command or Action	Purpose
<p>Step 3 <code>snmp-server host <i>host-addr</i> [traps informs] [version {1 2c 3 [auth noauth priv]]] [community-string] [udp-port <i>port</i>] [notification-type] [vrf <i>vrf-name</i>]</code></p> <p>Example:</p> <pre>Router(config)# snmp-server host 172.20.2.160 traps comaccess mpls-vpn</pre>	<p>Specifies the recipient of an SNMP notification operation.</p> <ul style="list-style-type: none"> The <i>host-addr</i> argument specifies the name or Internet address of the host (the targeted recipient). The traps keyword sends SNMP traps to this host. This is the default. The informs keyword sends SNMP informs to this host. The version keyword specifies the version of the SNMP used to send the traps. Version 3 is the most secure model, as it allows packet encryption with the priv keyword. If you use the version keyword, you must specify one of the following: <ul style="list-style-type: none"> 1--SNMPv1. This option is not available with informs. 2c --SNMPv2C. 3--SNMPv3. The following three optional keywords can follow the version 3 keyword (auth, noauth, priv). The <i>community-string</i> argument is a password-like community string sent with the notification operation. The udp-port <i>port</i> keyword argument pair names the UDP port of the host to use. The default is 162. The <i>notification-type</i> argument specifies the type of notification to be sent to the host. If no type is specified, all notifications are sent. The vrf <i>vrf-name</i> keyword argument pair specifies the VRF table that should be used to send SNMP notifications.
<p>Step 4 <code>snmp-server enable traps mpls vpn [illegal-label] [max-thresh-cleared] [max-threshold] [mid-threshold] [vrf-down] [vrf-up]</code></p> <p>Example:</p> <pre>Router(config)# snmp-server enable traps mpls vpn vrf- up vrf-down</pre>	<p>Enables the router to send MPLS VPN specific SNMP notifications (traps and informs).</p> <ul style="list-style-type: none"> The illegal-label keyword enables a notification for any illegal labels received on a VRF interface. Labels are illegal if they are outside the legal range, do not have an LFIB entry, or do not match table IDs for the label. The max-thresh-cleared keyword enables a notification when the number of routes falls below the limit after the maximum route limit was attempted. The max-threshold keyword enables a notification that a route creation attempt was unsuccessful because the maximum route limit was reached. Another mplsNumVrfRouteMaxThreshExceeded notification is not sent until the number of routes falls below the maximum threshold and reaches the maximum threshold again. The max-threshold value is determined by the maximum routes command in VRF configuration mode. The mid-threshold keyword enables a notification of a warning that the number of routes created has crossed the warning threshold. This warning is sent only at the time the warning threshold is exceeded. The vrf-down keyword enables a notification for the removal of a VRF from an interface or the transition of an interface to the down state. The vrf-up keyword enables a notification for the assignment VRF to an interface that is operational or for the transition of a VRF interface to the operationally up state.

Command or Action	Purpose
Step 5 end	(Optional) Exits to privileged EXEC mode.
Example: Router(config)# end	

Configuring Threshold Values for MPLS VPN--SNMP Notifications

Perform this task to configure the following threshold values for MPLS VPN--SNMP notifications:

- The **mplsNumVrfRouteMidThreshExceeded** notification event is generated and sent when the middle (warning) threshold is crossed. You can configure this threshold in the CLI by using the **maximum routes** command in VRF configuration mode. This notification is sent to the NMS only at the time the threshold is exceeded. Whenever the number of routes falls below this threshold and exceeds the threshold again, a notification is sent to the NMS.
- The **mplsNumVrfRouteMaxThreshExceeded** notification event is generated and sent when you attempt to create a route on a VRF that already contains the maximum number of routes as defined by the **maximum routes** command in VRF configuration mode. A trap notification is sent to the NMS when you attempt to exceed the maximum threshold. Another **mplsNumVrfRouteMaxThreshExceeded** notification is not sent until the number of routes falls below the maximum threshold and reaches the maximum threshold again.

See the figure above for an example of how this notification works and for a comparison of the maximum and warning thresholds.



Note

The **maximum routes** command sets the number of routes for a VRF. You *cannot* exceed the number of routes in the VRF that you set with the **maximum routes limit warn-threshold** command. Prior to this implementation of the PPVPN-MPLS-VPN MIB, you were not notified when this threshold (or the warning threshold) was reached.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf vrf-name**
4. **maximum routes limit {warn-threshold | warn-only}**
5. **end**

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 <code>enable</code></p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
<p>Step 2 <code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 <code>ip vrf vrf-name</code></p> <p>Example:</p> <pre>Router(config)# ip vrf vpn1</pre>	<p>Configures a VRF routing table and enters VRF configuration mode.</p> <ul style="list-style-type: none"> The <i>vrf-name</i> argument specifies the name assigned to a VRF.
<p>Step 4 <code>maximum routes limit {warn-threshold warn-only}</code></p> <p>Example:</p> <pre>Router(config-vrf)# maximum routes 10000 80</pre> <p>Example:</p> <pre>or</pre> <p>Example:</p> <pre>Router(config-vrf)# maximum routes 10000 warn-only</pre>	<p>Limits the maximum number of routes in a VRF to prevent a PE router from importing too many routes.</p> <ul style="list-style-type: none"> The <i>limit</i> argument specifies the maximum number of routes allowed in a VRF. The range is from 1 to 4,294,967,295. The <i>warn-threshold</i> argument generates a warning when the number of routes set by the <i>warn-threshold</i> argument is reached and rejects routes that exceed the maximum number set in the <i>limit</i> argument. The warning threshold is a percentage from 1 to 100 of the maximum number of routes specified in the <i>limit</i> argument. The warn-only keyword specifies that a SYSLOG error message is issued when the maximum number of routes allowed for a VRF exceeds the limit threshold. However, additional routes are still allowed.
<p>Step 5 <code>end</code></p> <p>Example:</p> <pre>Router(config-vrf)# end</pre>	<p>(Optional) Exits to privileged EXEC mode.</p>

Configuration Examples for PPVPN MPLS VPN MIB

- [Configuring the SNMP Community Examples, page 121](#)
- [Configuring the Router to Send SNMP Traps Example, page 121](#)
- [Configuring Threshold Values for MPLS VPN--SNMP Notifications Examples, page 121](#)

Configuring the SNMP Community Examples

The following example shows enabling a simple SNMP community group. This configuration permits any SNMP client to access all PPVPN-MPLS-VPN MIB objects with read-only access using the community string comaccess.

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

Verify that the SNMP master agent is enabled for the PPVPN MPLS VPN MIB feature:

```
Router# show running-config | include snmp-server
Building configuration...
....
snmp-server community comaccess RO
....
```



Note

If you do not see any “snmp-server” statements, SNMP is not enabled on the router.

Configuring the Router to Send SNMP Traps Example

The following example shows you how to enable the router to send MPLS VPN notifications to host 172.20.2.160 using the comaccess community string if a VRF transitions from an up or down state.

```
Router# configure terminal
Router(config)# snmp-server host 172.20.2.160 traps comaccess mpls-vpn
Router(config)# snmp-server enable traps mpls vpn vrf-up vrf-down
```

Configuring Threshold Values for MPLS VPN--SNMP Notifications Examples

The following example shows how to set a maximum threshold of 10000 routes and a warning threshold that is 80 percent of the maximum threshold for a VRF named vpn1 on a router:

```
Router(config)# ip vrf vpn1
Router(config-vrf)# maximum routes 10000 80
```

The following example shows how to set a warning threshold of 10000 routes for a VRF named vpn2 on a router. An error message is generated; however, additional routes are still allowed because a maximum route threshold is not set with this command.

```
Router(config)# ip vrf vpn2
Router(config-vrf)# maximum routes 10000 warn-only
```

Additional References

Related Documents

Related Topic	Document Title
Basic MPLS VPNs	Configuring MPLS Layer 3 VPNs
MPLS VPN Carrier Supporting Carrier	<ul style="list-style-type: none"> MPLS VPN Carrier Supporting Carrier Using LDP and an IGP MPLS VPN Carrier Supporting Carrier with BGP
MPLS VPN InterAutonomous Systems	MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

Standards

Standard	Title
<i>draft-ietf-ppvpn-mpls-vpn-mib-03</i>	<i>MPLS/BGP Virtual Private Network Management Information Base Using SMIPv2</i>

MIBs

MIB	MIBs Link
<ul style="list-style-type: none"> MPLS-VPN-MIB CISCO-IETF-PPVPN-MPLS-VPN-MIB 	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2233	<i>The Interfaces Group MIB using SMIPv2</i>
RFC 2547bis	<i>BGP/MPLS VPNs</i>

Technical Assistance

Description	Link
<p>The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.</p> <p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>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.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	<p>http://www.cisco.com/techsupport</p>

Feature Information for PPVPN MPLS VPN 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 27 Feature Information for Monitoring MPLS VPNs with MIBs

Feature Name	Releases	Feature Configuration Information
MPLS VPN--MIB Support	12.0(21)ST 12.0(22)S 12.2(13)S 12.2(15)T 12.0(24)S1 12.0(25)S 12.0(30)S	This feature allows you to monitor and manage MPLS VPNs using MIBs.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

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