



MPLS Enhancements to Interfaces MIB

Feature History

| Release | Modification |
|-----------|---|
| 12.0(23)S | This feature was introduced. |
| 12.2(18)S | This feature was integrated into Cisco IOS Release 12.2(18)S. |

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

This document includes the following major sections:

- [Feature Overview, page 1](#)
- [Supported Platforms, page 13](#)
- [Supported Standards, MIBs, and RFCs, page 13](#)
- [Prerequisites, page 14](#)
- [Configuration Tasks, page 14](#)
- [Configuration Examples, page 15](#)
- [Command Reference, page 16](#)
- [Glossary, page 19](#)

Feature Overview

The Interfaces MIB (IF MIB) provides a Simple Network Management Protocol (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 IOS.

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 new 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.



Corporate Headquarters:

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

Copyright © 2003 Cisco Systems, Inc. All rights reserved.

The enhancements also allow you to view 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 new interface is stacked above real Cisco IOS interfaces or subinterfaces, such as Ethernet (et0) or ATM (at1/1.1).

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

You can also create the interface layer if you enable MPLS traffic engineering (TE) by using the interface command **mpls traffic-eng tunnels**.

**Note**

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

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

MIB Tables

The IF MIB consists of the following tables:

- ifTable ([Table 1](#))—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.
- ifXTable ([Table 2](#))—Contains objects that have been added to the IF MIB as a result of the interface evolution effort or replacements for objects from the original MIB-II ifTable. This table also contains objects that were previously in the ifExtsTable.
- ifStackTable ([Table 3](#))—Contains objects that define the relationships among the sublayers of an interface.
- ifRcvAddressTable ([Table 4](#))—Contains objects that define the media-level addresses received by an interface.

**Note**

There are objects that are not supported by some interface types. The corresponding MPLS layer objects may not be supported if they depend on an unsupported object at the underlying layer.

The notation used in the IF MIB follows the conventions defined in Abstract System Notation One (ASN.1). ASN.1 defines an Open System Interconnection (OSI) language used to describe data types independently 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.

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

ifTable Objects

Table 1 lists the ifTable objects and their descriptions.

Table 1 *ifTable Objects and Descriptions*

| Objects | Description |
|---------------|--|
| ifIndex | A unique value, greater than zero, for each interface. |
| ifDescr | A textual string containing information about the MPLS layer. <underlying interface name>-mpls layer |
| ifType | The kind of interface. Additional values for ifType are assigned by the Internet Assigned Numbers Authority (IANA) through updating the syntax of the IANAifType textual convention. MPLS-layer entries are type mpls (166). |
| ifMtu | The size of the largest MPLS-encapsulated packet that can be sent or received on the interface, specified in octets. By default, this object is the same as the underlying physical or virtual interface, but can be configured by using the mpls mtu command. |
| ifSpeed | An estimate of the interface's current bandwidth in bits per second. This object is the same as the underlying physical or virtual interface; however, it has no MPLS-specific value. |
| ifPhysAddress | The interface's address at its protocol sublayer. This object is the same as the underlying physical or virtual interface. |
| ifAdminStatus | The administrative state of the underlying physical or virtual interface. This object is the same as the underlying physical or virtual interface. |
| ifOperStatus | The operational state of MPLS on the interface. Use the show mpls interface command to display this information. Note MPLS can be in the up or down state when the underlying physical or virtual interface is up. However, when the underlying physical or virtual interface is down, MPLS is always in the down state. |
| ifLastChange | The value of sysUpTime when the interface entered its current MPLS-specific operational state. |
| ifInOctets | The total number of MPLS-encapsulated octets received on the interface. |
| ifInUcastPkts | The number of MPLS-encapsulated packets received on the underlying physical or virtual interface. |

Table 1 *ifTable Objects and Descriptions (continued)*

| Objects | Description |
|-------------------|---|
| ifInNUcastPkts | <p>The number of packets, delivered by this sublayer to a higher sublayer, which were addressed to a multicast or broadcast address at this sublayer.</p> <p>This object is not supported on the MPLS layer because the event occurs on the underlying physical or virtual interface where it is counted.</p> |
| ifInDiscards | <p>The number of inbound packets selected to be discarded even though no errors had been detected to prevent delivery to a higher-layer protocol. One reason for discarding such a packet is to free up buffer space.</p> <p>This object is not supported on the MPLS layer because the event occurs on the underlying physical or virtual interface where it is counted.</p> |
| ifInErrors | <p>The number of inbound packets that contained errors preventing them from being delivered to a higher-layer protocol.</p> <p>This object is not supported on the MPLS layer because the event occurs on the underlying physical or virtual interface where it is counted.</p> |
| ifInUnknownProtos | <p>The number of packets that were discarded because of an unknown or unsupported protocol received via the interface.</p> <p>This object is not supported on the MPLS layer because the event occurs on the underlying physical or virtual interface where it is counted.</p> |
| ifOutOctets | The total number of MPLS-encapsulated octets transmitted from the interface. |
| ifOutUcastPkts | The number of MPLS-encapsulated packets transmitted from the underlying physical or virtual interface. |
| ifOutNUcastPkts | <p>The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a multicast or broadcast address at this sublayer, including those that were discarded or not sent.</p> <p>This object is not supported on the MPLS layer because the event occurs on the underlying physical or virtual interface where it is counted.</p> |
| ifOutDiscards | <p>The number of outbound packets selected to be discarded even though no errors had been detected to prevent their transmission. One reason for discarding such a packet is to free up buffer space.</p> <p>This object is not supported on the MPLS layer because the event occurs on the underlying physical or virtual interface where it is counted.</p> |

Table 1 *ifTable Objects and Descriptions (continued)*

| Objects | Description |
|-------------|---|
| ifOutErrors | The number of outbound packets that could not be transmitted because of errors. This object is not supported on the MPLS layer because the event occurs on the underlying physical or virtual interface where it is counted. |
| ifOutQLen | The length of the output packet queue (in packets). This object is the same as the underlying physical or virtual interface. It is not supported for MPLS. |
| ifSpecific | A reference to MIB definitions specific to the particular media being used to recognize the interface. This object is not supported because there is no such reference for MPLS. |

ifXTable Objects

Table 2 lists the ifXTable objects and their descriptions.



Note

High-capacity (HC) objects supported in the IF MIB require SNMPv2c or higher.

Table 2 *ifXTable Objects and Descriptions*

| Object | Definition |
|--------------------|---|
| ifName | The textual name of the underlying physical or virtual interface. |
| ifInMulticastPkts | The number of packets, which were addressed to a multicast address at this sublayer and delivered by this sublayer to a higher sublayer. This object is not supported because MPLS currently does not support multicast. |
| ifInBroadcastPkts | The number of packets, which were addressed to a broadcast address at this sublayer and delivered by this sublayer to a higher sublayer. This object is not supported because MPLS currently does not support broadcast. |
| ifOutMulticastPkts | The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a multicast address at this sublayer, including those that were discarded or not sent. This object is not supported because MPLS currently does not support multicast. |

Table 2 *ifXTable Objects and Descriptions (continued)*

| Object | Definition |
|---------------------|--|
| ifOutBroadcastPkts | <p>The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a broadcast address at this sublayer, including those that were discarded or not sent.</p> <p>This object is not supported because MPLS currently does not support broadcast.</p> |
| ifHCInOctets | <p>The total number of MPLS-encapsulated octets received on the interface.</p> <p>This object is a 64-bit version of ifInOctets.</p> |
| ifHCInUcastPkts | <p>The number of MPLS-encapsulated packets received on the underlying physical or virtual interface.</p> <p>This object is a 64-bit version of ifInUcastPkts.</p> |
| ifHCInMulticastPkts | <p>The number of packets, which were addressed to a multicast address at this sublayer and delivered by this sublayer to a higher sublayer.</p> <p>This object is not supported because MPLS currently does not support multicast.</p> <p>This object is a 64-bit version of ifInMulticastPkts.</p> |
| ifHCInBroadcastPkts | <p>The number of packets, which were addressed to a broadcast address at this sublayer and delivered by this sublayer to a higher sublayer.</p> <p>This object is not supported because MPLS currently does not support broadcast.</p> <p>This object is a 64-bit version of ifInBroadcastPkts.</p> |
| ifHCOctets | <p>The total number of MPLS-encapsulated octets transmitted from the interface.</p> <p>This object is a 64-bit version of ifOutOctets.</p> |
| ifHCOUcastPkts | <p>The number of MPLS-encapsulated packets transmitted from the underlying physical or virtual interface.</p> <p>This object is a 64-bit version of ifOutUcastPkts.</p> |
| ifHCOMulticastPkts | <p>The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a multicast address at this sublayer, including those that were discarded or not sent.</p> <p>This object is not supported because MPLS currently does not support multicast.</p> <p>This object is a 64-bit version of ifOutMulticastPkts.</p> |
| ifHCOBroadcastPkts | <p>The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a broadcast address at this sublayer, including those that were discarded or not sent.</p> <p>This object is not supported because MPLS currently does not support broadcast.</p> <p>This object is a 64-bit version of ifOutBroadcastPkts.</p> |

Table 2 *ifXTable Objects and Descriptions (continued)*

| Object | Definition |
|----------------------------|--|
| ifLinkUpDownTrapEnable | Indicates whether LinkUp/LinkDown traps should be generated for this interface. Link up/down traps are not supported in this release; consequently, this object is set to disabled(2) for MPLS. |
| ifHighSpeed | An estimate of the interface's current bandwidth in units of 1,000,000 bits per second. This object is the same as the underlying physical or virtual interface; however, it has no MPLS-specific value. |
| ifPromiscuousMode | Has a value of false(2) if this interface accepts only packets and frames that are addressed to this station; has a value of true(1) when the station accepts all packets and frames transmitted on the media. This object is the same as the underlying physical or virtual interface. |
| ifConnectorPresent | Has the value true(1) if the interface sublayer has a physical connector and the value false(2) otherwise. For MPLS, the value is always false(2). |
| ifAlias | Another name for an interface as specified by a network manager. This object is the same as the underlying physical or virtual interface. |
| ifCounterDiscontinuityTime | The value of sysUpTime on the most recent occasion at which any one or more of this interface's MPLS-specific counters experienced an interruption in incrementing. |

ifStackTable Objects

[Table 3](#) lists the ifStackTable objects and their descriptions.

Table 3 *ifStackTable Objects and Definitions*

| Object | Definition |
|--------------------|--|
| ifStackHigherLayer | <p>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.</p> <p>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.</p> |
| ifStackLowerLayer | <p>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.</p> <p>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.</p> |
| ifStackStatus | Used to create and delete rows in the ifStackTable; status is always active(1) for MPLS. |

ifRcvAddressTable Objects

Table 4 lists the ifRcvAddressTable objects and their descriptions.


Note

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

Table 4 *ifRcvAddressTable Objects and Descriptions*

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

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.

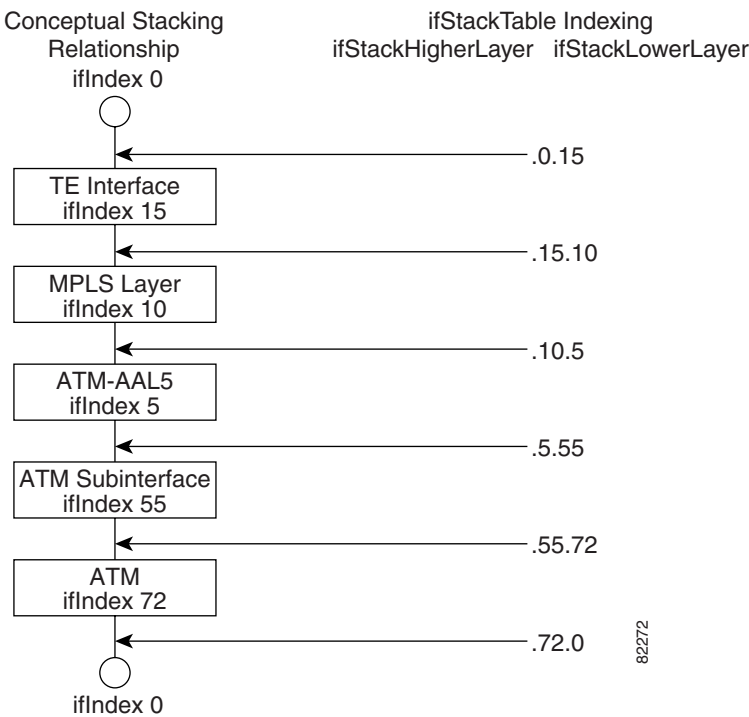
- ifTableLastChange—The value of sysUpTime at the time of the last creation or deletion of an entry in the ifTable. If the number of entries has been unchanged since the last reinitialization of the local network management subsystem, then this object contains a zero value.

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 [Figure 1](#), 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 [Figure 1](#), 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



[Table 5](#) describes the indexing of the ifStackTable for the layer relationships shown in [Figure 1](#).



Note

The order of the entries in [Table 5](#) may not be the same as that seen in the MIB walk, which has to follow SNMP ordering rules.

Table 5 **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 |

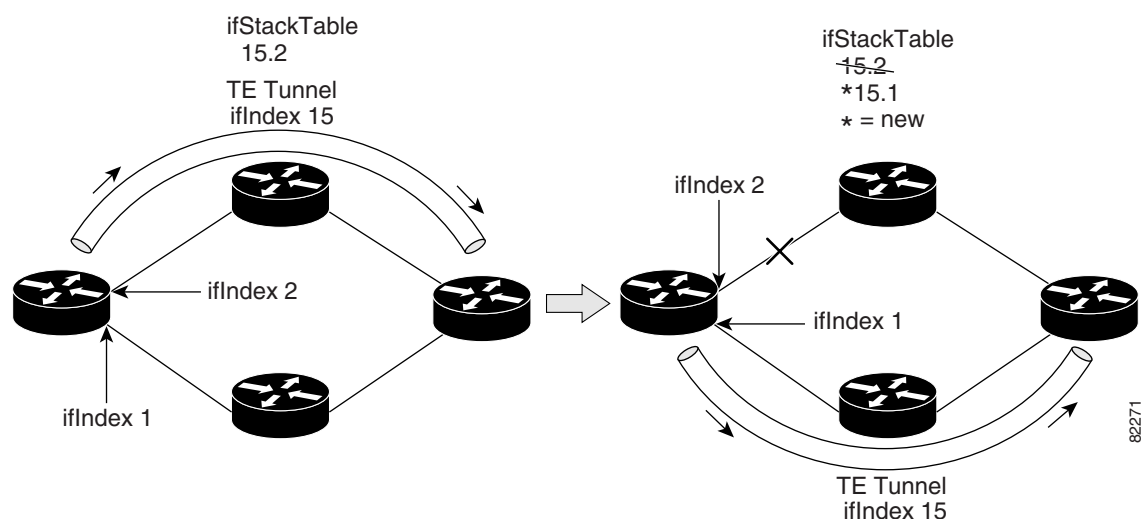
Stacking Relationships for Traffic Engineering Tunnels

MPLS traffic engineering (TE) tunnels are represented in Cisco IOS 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 as well as 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.

Figure 2 shows a TE tunnel before (left) and after (right) being rerouted and the affect on the ifStackTable. When ifIndex 2 fails, the TE tunnel is rerouted through ifIndex 1, the 15.2 entry is removed from the ifStackTable, and the 15.1 entry is added.

Figure 2 **Sample TE Tunnel Stacking Relationship**

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.

Table 6 shows the specific changes.

Table 6 *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 packets
- mplsInterfaceInPackets—Count only MPLS-encapsulated packets

Benefits

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

Restrictions

- 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.

Related Documents

Cisco Documentation

For information on configuring SNMP using Cisco IOS software, see the following documents:

- [Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2](#)
- [Cisco IOS Configuration Fundamentals Command Reference, Release 12.2](#)
- [MPLS Label Switching Router MIB, Cisco IOS Release 12.0\(14\)ST feature module](#)

Other Documentation

- *Multiprotocol Label Switching (MPLS) Label Switch Router (LSR) Management Information Base*, Internet draft, January 2002 [draft-ietf-mpls-lsr-mib-08.txt]; Srinivasan, C., Viswanathan, A., and Nadeau, T.D.

For information on using SNMP MIB features, see the appropriate documentation for your network management system.

Supported Platforms

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

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

Interfaces Group MIB (IF MIB)

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

<http://tools.cisco.com/ITDIT/MIBS/servlet/index>

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

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

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

<http://www.cisco.com/register>

RFCs

- RFC 2233, *The Interfaces Group MIB using SMIPv2*
- RFC 1229, *Extensions to the Generic-Interface MIB*
- RFC 1213, *Management Information Base for Network Management of TCP/IP-based internets: MIB-II*
- RFC 1157, *A Simple Network Management Protocol (SNMP)*
- RFC 1156, *Management Information Base for Network Management of TCP/IP-based internets*

Prerequisites

- The MPLS Enhancements to Interfaces MIB feature requires the following:
- SNMP must be installed and enabled on the label switch routers (LSRs)
 - MPLS must be enabled on the LSRs
 - MPLS IP must be enabled on an interface or an MPLS TE tunnel enabled on an interface

Configuration Tasks

- See the following sections for configuration tasks for the MPLS Enhancements to Interfaces MIB feature. Each task in the list is identified as either optional or required.
- [Enabling the SNMP Agent](#) (required)
 - [Verifying That the SNMP Agent Has Been Enabled](#) (optional)

Enabling the SNMP Agent

The SNMP agent for the IF MIB is disabled by default. To enable the SNMP agent, use the following commands:

| | Command | Purpose |
|--------|--|--|
| Step 1 | Prompt# telnet XXX.XXX.XXX.XXX | Accesses the router identified by the specified IP address (represented by xxx.xxx.xxx.xxx) through a Telnet session. |
| Step 2 | Router# enable | Enters privileged mode. |
| Step 3 | Router# show running-configuration | Displays the running configuration so that you can determine if an SNMP agent is already running. If no SNMP information is displayed, continue with the steps below. If any SNMP commands are listed, you can modify them or leave them as they are. |
| Step 4 | Router# configure terminal | Enters configuration mode. |
| Step 5 | Router(config)# snmp-server community xxxxxx RO | Enables the read-only community string, where xxxxxx is the read-only community string. |
| Step 6 | Router(config)# exit | Exits configuration mode and returns to the main prompt. |
| Step 7 | Router# write memory | Writes the modified configuration to NVRAM so that the settings stay permanently. |

Verifying That the SNMP Agent Has Been Enabled

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

Step 1 Access the router through a Telnet session:

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

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

Step 2 Enter privileged mode:

```
Router# enable
```

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

```
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

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
```

Command Reference

This section documents new or modified commands. All other commands used with this feature are documented in the Cisco IOS Release 12.2 command reference publications.

- [snmp-server community](#)

snmp-server community

To configure read-only or read/write Simple Network Management Protocol (SNMP) community strings, use the **snmp-server community** command in global configuration mode. To change the community string to its default value, use the **no** form of this command.

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

no snmp-server community *string*

Syntax Description

| | |
|------------------------------|--|
| <i>string</i> | Community string of 1 to 32 alphanumeric characters. The community string acts like a password and permits access to the SNMP protocol. No blank character spaces are allowed. |
| view <i>view-name</i> | (Optional) Name of a previously defined view. The view defines the objects available to the community. |
| ro | (Optional) Configures read-only access. Authorized management stations can only retrieve MIB objects. |
| rw | (Optional) Configures read/write access. Authorized management stations can retrieve and modify MIB objects. |
| <i>number</i> | (Optional) 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 SNMPv1 agent. |

Defaults

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

Command Modes

Global configuration

Command History

| Release | Modification |
|------------|--|
| 10.0 | This command was introduced. |
| 12.0(14)ST | This command was integrated into Cisco IOS Release 12.0(14)ST. |
| 12.2(2)T | This command was integrated into Cisco IOS Release 12.2(2)T. |
| 12.0(23)S | This command was integrated into Cisco IOS Release 12.0(23)S. |
| 12.2(18)S | This command was integrated into Cisco IOS Release 12.2(18)S. |

Usage Guidelines

The **no snmp-server** command disables both versions of SNMP (SNMPv1 and SNMPv2).

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

Examples

In this example, the read/write community string is set to newstring:

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

In the following example, the string comaccess is assigned to SNMPv1, allowing read-only access. IP access list 4 is enabled to use the community string:

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

In the following example, the string mgr is assigned to SNMPv1, allowing read/write access to the objects in the restricted view:

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

In the following example, the community comaccess is removed:

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

In the following example, both versions of SNMP are disabled:

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

Related Commands

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

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—The forwarding of IP (or other network layer) packets by 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 switch router. A device that forwards 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 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 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.

MPLS interface—An interface on which MPLS traffic is enabled.

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

NMS—network management system. A 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 an 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 an 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 an 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.

CCVP, the Cisco logo, and Welcome to the Human Network are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn is a service mark of Cisco Systems, Inc.; and Access Registrar, Aironet, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, LightStream, Linksys, MeetingPlace, MGX, Networkers, Networking Academy, Network Registrar, PIX, ProConnect, ScriptShare, SMARTnet, StackWise, The Fastest Way to Increase Your Internet Quotient, and TransPath are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website 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. (0711R)

Copyright © 2003 Cisco Systems, Inc. All rights reserved.