Cisco CRS and Cisco XR 12000 Series Router MIB Specifications Guide

Cisco IOS XR Software

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

Preface

This guide describes the implementation of the Simple Network Management Protocol (SNMP) and MIB for the Cisco Carrier Routing System (CRS) and Cisco XR 12000 Series Router executing IOS XR. For the Cisco XR 12000 Series Router executing IOS, see the IOS MIB Locator Tool at http://tools.cisco.com/ITDIT/MIBS/servlet/index. SNMP provides a set of commands for setting and retrieving the values of operating parameters on the Cisco CRS and Cisco XR 12000 Series routers. The router information is stored in a virtual storage area called a MIB, which contains many MIB objects that describe router components and provides information about the status of the components.

This preface provides an overview of this guide with the following sections:

- Revision History
- Audience
- Organization
- Terminology and Definitions
- Obtaining Documentation and Submitting a Service Request

Revision History

The following Revision History tables record technical changes, additions, and corrections to this document. The table shows the release number and document revision number for the change, the date of the change, and a summary of the change.

Cisco IOS XR Release	Part Number	Publication Date
IOS XR Software Release 5.2.0	OL-29005-03	July 2014
IOS XR Software Release 5.1.2	OL-29005-02	April 2014
IOS XR Software Release 4.3	OL-29005-01	February 2013

Description of Changes

- Updates made to these sections:
  - Updated BGP4-MIB Tables and Descriptions Table
  - Updated BGP4-MIB Constraints Table
  - Updated CISCO-CDP-MIB Tables and Descriptions Table
  - Updated CISCO-CLASS-BASED-QOS-MIB Constraints Table
  - Updated CISCO-IETF-BFD-MIB Constraints Table
  - Updated CISCO-IETF-FRR-MIB Constraints Table
  - Updated CISCO-IETF-IPMROUTE-MIB Constraints Table
  - Updated CISCO-RTTMON-MIB Tables and Descriptions Table

- Updated CISCO-RTTMON-MIB Constraints Table
- Updated CISCO-SYSLOG-MIB Constraints Table
- Updated IEEE8023-LAG-MIB Constraints Table
- Updated IPV6-MIB Tables and Descriptions Table
- Updated ISIS-MIB Constraints Table
- Updated MPLS-L3VPN-STD-MIB Constraints Table
- Updated MPLS-LDP-GENERIC-STD-MIB
- Updated MPLS-TC-STD-MIB
- Updated MPLS-TE-STD-MIB
- Updated MPLS-TE-STD-MIB Constraints
- Updated RSVP-MIB Constraints

Audience

This guide is intended for system and network administrators who configure the Cisco CRS or Cisco XR 12000 Series router for operation and monitor performance in the network.

This guide may also be useful for application developers who are developing management applications for the Cisco CRS or Cisco XR 12000 Series router.

Organization

This guide contains the following chapters:

Chapter	Description
Chapter 1, "Cisco CRS and Cisco XR 12000 Series Router MIB Overview"	Provides background information about SNMP and its implementation on the Cisco CRS and Cisco XR 12000 Series routers.
Chapter 2, "Configuring MIB Support"	Provides instructions for configuring SNMP management support on the Cisco CRS and Cisco XR 12000 Series routers.
Chapter 3, "Platform-Independent MIB Specifications"	Describes MIBs for which the majority of their operation and data is independent of the specific platform or hardware or the feature is supported across all XR platforms.
Chapter 4, "Cisco XR 12000 Series Router MIB Specifications"	Describes each MIB included on the Cisco CRS. Each description lists any constraints as to how the MIB is implemented on the router.
Chapter 5, "Cisco XR 12000 Series Router	Describes each MIB included on the Cisco XR 12000

MIB Specifications”	Series router. Each description lists any constraints as to how the MIB is implemented on the router.
Chapter 6 “Monitoring Notifications”	Describes the SNMP notifications supported by the Cisco CRS and Cisco XR 12000 Series routers, provides a description of each notification, a probable cause, and recommended action to take.
Appendix 1, “Using MIBs”	Provides information about how to use SNMP to perform system functions such as bulk-file retrieval and QoS[1].
Appendix 2, “QoS MIB Implementation”	Provides information about how to implement QoS in addition to a matrix that defines which objects support QoS policy actions.
Appendix 3, “RFC 1213”	Provides information about the evolution of RFC1213 to other MIBs.
Appendix 4, “Process Information for SNMP-centric MIBs”	Provides process information for SNMP-centric MIBs.
Appendix 5, “IOS XR SNMP Best Practices”	Provides information on best practices to be adopted by an OSS[2] for optimized use of IOS-XR SNMP protocol.

[1] QoS = Quality of Service

[2] OSS = Operations Support System

Terminology and Definitions

This section discusses conventions and terminology used in this guide.

- Alarm–In SNMP, the word *alarm* is commonly misused to mean the same as a trap (see the Trap definition below). *Alarm* represents a condition which causes an SNMP trap to be generated.

Note Many commands use the word **traps** in the command syntax. Unless there is an option in the command to select traps. Use the **snmp-server host** and **snmp-server** *notification* command to specify whether to send SNMP notifications as traps.

- Element Management System (EMS)–EMS manages a specific portion of the network. For example, the SunNet Manager, an SNMP management application, is used to manage SNMP-manageable elements. Element Managers may manage asynchronous lines, multiplexers, Private Automatic Branch Extension (PABX), proprietary systems, or applications.
- Management Information Base (MIB)–Management objects available in an SNMP managed device. The information is represented in Abstract Syntax Notation 1 (ASN.1). This is a way of logically grouping data so that it is easily understood by all.
- MIB-II–Successor to MIB-I, which was the original standard SNMP MIB.
- Multiprotocol Label Switching (MPLS)–Standardized version of the Cisco original tag-switching proposal. It uses a label-forwarding paradigm (forward packets based on labels).
- SNMP–Application layer protocol that allows you to remotely manage networked devices. The *simple* in SNMP is only in contrast to protocols that are thought to be even more complex than SNMP. SNMP consists of the following components: a management protocol, a definition of management information and events, a core set of management information and events, and a mechanism and approach used to manage the use of the protocol including security and access control.

- Trap—Device-initiated SNMP notification message. The contents of the message might be simply informational, but it is mostly used to report real-time trap information. Traps can be used in conjunction with other SNMP mechanisms, as in trap-directed polling.
- User Datagram Protocol (UDP)—Connectionless, non-reliable IP-based transport protocol.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at: http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as an RSS feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service. Cisco currently supports RSS Version 2.0.

Cisco CRS and Cisco XR 12000 Series Router MIB Overview

This chapter provides an overview of the Cisco Carrier Routing System and Cisco XR 12000 Series Router. This chapter contains the following topics:

- Benefits of MIB Enhancements
- SNMP Overview
- Object Identifiers

Benefits of MIB Enhancements

The Cisco Carrier Routing System router management feature and the Cisco XR 12000 Series router management feature allow the routers to be managed through the Simple Network Management Protocol (SNMP).

Use the CRS or Cisco XR 12000 Series router management feature to:

- Manage and monitor the resources through an SNMP-based Network Management System (NMS).
- Use SNMP **set** and **get** requests to access information in the router MIBs.
- Reduce the amount of time and system resources required to perform functions such as inventory management.

Other benefits include:

- A standards-based technology (SNMP) for monitoring faults and performance on the router
- Support for all SNMP versions (SNMPv1, SNMPv2c, and SNMPv3)
- Notification of faults, alarms, and conditions that might affect services
- A way to access router information other than through the Command-Line Interface (CLI) or Extensible Markup Language (XML)

SNMP Overview

The *SNMP* is an application-layer protocol that provides a standardized framework and a common language used for monitoring and managing devices in a network.

The SNMP framework has three parts:

- SNMP manager—System used to control and monitor the activities of network hosts using SNMP. The most common managing system is called a *NMS*. The term NMS can be applied to either a dedicated device used for network management, or the applications used on a network management device. A variety of network management applications are available for use with SNMP. These features range from simple command-line applications to feature-rich graphical user interfaces (such as the CiscoWorks2000 products).
- SNMP agent—Software component in a managed device that maintains the data for the device and reports the data, as needed, to managing systems. The agent and MIB reside in the routing device (router, access server, or switch). To enable the SNMP agent on a managed device, you must define the relationship between the manager and the agent (see the “Enabling SNMP Support” section).
- Management Information Base (MIB – Database of objects that can be managed on a device. This database describes various components and provides information about the attributes of the components of a network device.

Instead of defining a large set of commands, SNMP places all operations in a get-request, get-next-request, and set-request format. For example, an SNMP manager can get a value from an SNMP agent or set a value in that SNMP agent.

MIB Description

A MIB is a database of the objects that can be managed on a device. The managed objects or variables can be set or read to provide information on the network devices and interfaces and are organized hierarchically. The MIB consists of collections of managed objects identified by object identifiers. MIBs are accessed using a network management protocol such as SNMP. A managed object (sometimes called a *MIB object* or an *object*) is one of several characteristics of a managed device, such as a router. Managed objects comprise one or more object instances, which are essentially variables. The Cisco implementation of SNMP uses the definitions of MIB II variables described in RFC 1213.

MIBs contain two types of managed objects:

- Scalar objects—Define a single object instance (for example, ifNumber in the IF-MIB and bgpVersion in the BGP4-MIB).
- Columnar objects—Define multiple related objects such as zero, one, or more instances at any point in time that are grouped together in MIB tables (for example, ifTable in the IF-MIB defines the interface).
- System MIB variables are accessible through SNMP as follows:
- Accessing a MIB variable—Function is initiated by the SNMP agent in response to a request from the NMS. The agent retrieves the value of the requested MIB variable and responds to the NMS with that value.
- Setting a MIB variable—Function is initiated by the SNMP agent in response to a message from the NMS. The SNMP agent changes the value of the MIB variable to the value requested by the NMS.

SNMP Notifications

An SNMP agent can notify the SNMP manager when important system events occur, such as the following:

- Interface or card starts or stops running
- Temperature thresholds are crossed
- Authentication failures occur

When an agent detects an alarm condition, the agent:

- Logs information about the time, type, and severity of the condition
- Generates a notification message, which it then sends to a designated IP host

SNMP notifications are sent as Traps. Traps are unreliable messages, which do not require receipt acknowledgment from the SNMP manager.

The Cisco implementation of SNMP uses the definitions of SNMP traps described in RFC 1215.

When an agent detects an alarm condition, it logs information about the time, type, and severity of the condition and generates a notification message, which it then sends to a designated IP host. SNMP notifications are sent as *traps*. See the “Enabling Notifications” section for instructions on how to enable notifications and traps on the Cisco Carrier Routing System and Cisco XR 12000 Series Router. Use the **snmp-server host** command to specify that SNMP notifications are sent as traps. See “Monitoring Notifications” for information about Cisco Carrier Routing System and Cisco XR 12000 Series Router traps.

SNMP Versions

Cisco IOS XR Software supports the following SNMP versions:

- SNMPv1—Simple Network Management Protocol. Internet standard, defined in RFC 1157. Security is based on community strings.
- SNMPv2c—Community-string based administrative framework for SNMPv2. SNMPv2c is an update of the protocol operations and data types of SNMPv2p (SNMPv2 classic).
- SNMPv3—Version 3 of SNMP. SNMPv3 uses the following security features to provide secure access to devices:
    - Message integrity—Ensuring that a packet has not been tampered with in transit.
    - Authentication—Determining that the message is from a valid source.
    - Encryption—Scrambling the contents of a packet to prevent it from being learned by an unauthorized source.

SNMPv1 and SNMPv2c

Both SNMPv1 and SNMPv2c use a community-based form of security. The community of managers who are able to access the agent MIB is defined by an IP address access control list and password.

SNMPv2c support includes a bulk-retrieval mechanism and a more detailed error message reporting to management stations. The bulk-retrieval mechanism supports retrieval of tables and large quantities of information, minimizing the number of round-trip transmissions required. The improved SNMPv2c error-handling support includes expanded error codes that distinguish different kinds of error conditions; these conditions are reported through a single error code in SNMPv1. Error return codes report the error type. Three kinds of exceptions are also reported:

- No such object
- No such instance
- End of MIB view

SNMPv3

SNMPv3 provides security models and security levels:

- Security *model* is an authentication strategy that is set up for a user and the group in which the user resides.
- Security *level* is the permitted level of security within a security model.
- Combination of a security model and a security level determines the security mechanism employed when handling an SNMP packet.

SNMP Security Models and Levels

This table describes the security models and levels provided by the different SNMP versions.

Table 1-1 SNMP Security Models and Levels

Model	Level	Authentication	Encryption	Description

v1	noAuthNoPriv	Community string	No	Uses match on community string for authentication.
v2c	noAuthNoPriv	Community string	No	Uses match on community string for authentication.
v3	noAuthNoPriv	User name	No	Uses match on user name for authentication.
	authNoPriv	MD5 or SHA	No	Provides authentication based on HMAC-MD5 or HMAC-SHA algorithm.
	authPriv	MD5 or SHA	DES	Provides authentication based on HMAC-MD5 or HMAC-SHA algorithm. Also provides DES 56-bit encryption based on CBC-DES (DES-56) standard.

You must configure the SNMP agent to use the version of SNMP supported by the management station. An agent can communicate with multiple managers; for this reason, you can configure the Cisco IOS XR Software to support communications with one management station using the SNMPv1 protocol, one using the SNMPv2c protocol, and another using SMNPv3.

Requests For Comments

MIB modules are typically defined in Request for Comment (RFC) documents that have been submitted to the IETF for formal discussion and approval. RFCs are written by individuals or groups for consideration by the Internet Society and the Internet community as a whole.

Before getting an RFC status, recommendations are published as Internet Draft (I-D) documents. RFCs that have become recommended standards are also labeled as standards (STD) documents. For more information, see the Internet Society and IETF websites (http://www.isoc.org and http://www.ietf.org).

We provide private MIB extensions with each Cisco system. Cisco enterprise MIBs comply with the guidelines described in the relevant RFCs unless otherwise noted in the documentation.

Object Identifiers

An object identifier (OID) uniquely identifies a MIB object on a managed network device. The OID identifies the MIB object location in the MIB hierarchy, and provides a means of accessing the MIB object in a network of managed devices:

- Standard RFC MIB OIDs are assigned by the Internet Assigned Numbers Authority (IANA).
- Enterprise MIB OIDs are assigned by Cisco Assigned Numbers Authority (CANA).

Each number in the OID corresponds to a level of MIB hierarchy. For example, the OID 1.3.6.1.4.1.9.9.xyz represents the *.xyz* with the location in the MIB hierarchy as follows. Note that the numbers in parentheses are included to help show correspondence to the MIB hierarchy. In actual use, OIDs are represented as numerical values only.

iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).cisco(9).ciscoMgt(9).*nn*-MIB

You can uniquely identify a managed object, such as ifNumber in the IF-MIB, by its object name (iso.org.dod.internet.mgmt.enterprises.interfaces.ifNumber) or by its OID (1.3.6.1.2.1.2.1).

For a list of OIDs assigned to MIB objects, go to the following URL:

ftp://ftp.cisco.com/pub/mibs/oid/

SNMP Configuration Information

The following URLs provide information about configuring SNMP:

- http://www.cisco.com/en/US/docs/ios_xr_sw/iosxr_r3.7/system_management/configuration/guide/yc37snmp.html provides general information about configuring and implementing SNMP support. It is part of *Cisco IOS XR System Management Configuration Guide, Release 3.7*.
- http://www.cisco.com/en/US/docs/ios_xr_sw/iosxr_r3.7/system_management/command/reference/yr37snmp.html provides information about SNMP commands. It is part of *Cisco IOS XR System Management Command Reference, Release 3.7*.

Configuring MIB Support

This chapter describes how to configure SNMP (Simple Network Management Protocol) and MIB (Management Information Base) support for the Cisco Carrier Routing System and Cisco XR 12000 Series Router. It includes the following sections:

- Downloading and Compiling MIBs
- Enabling SNMP Support

Downloading and Compiling MIBs

The following sections provide information about how to download and compile MIBs for the Cisco Carrier Routing System and Cisco XR 12000 Series Router:

- Considerations for Working with MIBs
- Downloading MIBs
- Compiling MIBs

Considerations for Working with MIBs

While working with MIBs, consider the following:

- Mismatches on datatype definitions might cause compiler errors or warning messages. Although Cisco MIB datatype definitions are not mismatched, some standard RFC MIBs do mismatch, see the following example:

  ```
  MIB A defines: SomeDatatype ::= INTEGER(0..100)
  MIB B defines: SomeDatatype ::= INTEGER(1..50)
  ```

  This example is considered to be a trivial error and the MIB loads successfully with a warning message.

  The following example is considered as a nontrivial error (although the two definitions are essentially equivalent), and the MIB is not successfully parsed:

  ```
  MIB A defines: SomeDatatype ::= DisplayString
  MIB B defines: SomeDatatype ::= OCTET STRING (SIZE(0..255))
  ```

  If your MIB compiler treats these as errors, or to delete the warning messages, edit one of the MIBs that defines this same datatype so that the definitions match.

- Many MIBs import definitions from other MIBs. If your management application requires MIBs to be loaded, and you experience problems with undefined objects, consider loading the following MIBs in this order:

  SNMPv2-SMI.my

  SNMPv2-TC.my

  SNMPv2-MIB.my

  IF-MIB.my

  CISCO-SMI.my

- For information about how to download and compile Cisco MIBs, see: http://www.cisco.com/en/US/tech/tk648/tk362/technologies_tech_note09186a00800b4cee.shtml

Downloading MIBs

To download the MIBs onto your system, perform the following steps:

Step 1 Review the considerations, see the “Considerations for Working with MIBs” section.

Step 2 Go to one of the following Cisco URLs.

ftp://ftp.cisco.com/pub/mibs/v2

ftp://ftp.cisco.com/pub/mibs/v1

If the MIB is not at these URLs, go to one of the URLs in Step 5.

Step 3 Click the link for a MIB to download that MIB to your system.

Step 4 Select **File > Save** or **File > Save As** to save the MIB on your system.

Step 5 Download industry-standard MIBs from the following URLs:

http://www.ietf.org/

http://www.ipmplsforum.org/

Compiling MIBs

If you plan to integrate the Cisco Carrier Routing System or Cisco XR 12000 Series Router with an SNMP-based management application, you must compile the MIBs for that platform. For example, if you are running HP OpenView on a UNIX operating system, you must compile Cisco Carrier Routing System MIBs with the HP OpenView Network Management System (NMS).

Enabling SNMP Support

The following procedure summarizes how to configure the Cisco Carrier Routing System or the Cisco XR 12000 Series Router for SNMP support.

For detailed information about SNMP commands, go to the following URL:

- http://www.cisco.com/en/US/docs/ios_xr_sw/iosxr_r3.7/system_management/configuration/guide/yc37snmp.html provides general information about configuring and implementing SNMP support. It is part of *Cisco IOS XR System Management Configuration Guide, Release 3.7*.
- http://www.cisco.com/en/US/docs/ios_xr_sw/iosxr_r3.7/system_management/command/reference/yr37snmp.html provides information about SNMP commands. It is part of *Cisco IOS XR System Management Command Reference, Release 3.7*.

To configure the Cisco Carrier Routing System or Cisco XR 12000 Series Router for SNMP support, perform the following steps:

Step 1 Set up your basic SNMP configuration through the command-line interface (CLI) on the router. Note that these basic configuration commands are issued for SNMPv2c. For SNMPv3, you must also set up SNMP users and groups. (See the preceding list of documents for command and setup information.)

a. Define SNMP read-only and read-write communities:

```
Router (config)# snmp-server community Read_Only_Community_Name ro SystemOwner
Router (config)# snmp-server community Read_Write_Community_Name rw SystemOwner
```

b. Configure SNMP views (to limit the range of objects accessible to different SNMP user groups):

```
Router (config)# snmp-server view view_name oid-tree {included | excluded}
```

Step 2 Identify (by IP address) the host to receive SNMP notifications from the router:

```
Router (config)# snmp-server host host
```

Step 3 Configure the router to generate notifications. Use keywords to limit the number and types of messages generated.

```
Router (config)# snmp-server traps [notification-type] [notification-option]
```

For information about how to configure SNMP community strings, see the following URL:

http://www.cisco.com/c/en/us/td/docs/ios_xr_sw/iosxr_r3-6/system_management/command/reference/yr36/yr36snmp.html#wp1013028

Platform-Independent MIB Specifications

This chapter describes MIBs in which the majority of their operation and data is independent of the specific platform or hardware or the feature is supported across all IOS XR platforms. See Chapter 4, “Cisco Carrier Routing System MIBs” for more information on CRS MIBs and Chapter 5, “Cisco XR 12000 Series Router MIBs” for more information on Cisco XR 12000 Series router MIBs.

Each MIB description lists any constraints on how the MIB or its object identifiers (OIDs) are implemented on the platforms.

Unless noted otherwise, the implementation of a MIB follows the standard MIB that has been defined. Any MIB table or object not listed in the table is implemented as defined in the standard MIB definition.

This chapter contains the following sections:

- Platform-Independent MIBs
- Platform-Independent MIB Categories
- MIB Support Category Summary
- MIB Version String Description

Platform-Independent MIBs

Each MIB description lists relevant constraints about the implementation of the MIB on the IOS XR platforms. Any objects not listed in a table are implemented as defined in the MIB. For detailed MIB descriptions, see the standard MIB.

Note Not all MIBs included in a Cisco IOS XR Software release are fully supported by the router. Some MIBs are not supported at all. Other MIBs might work, but they have not been tested on the router. In addition, some MIBs are deprecated but cannot be removed from the software. When a MIB is included in the image, this does not necessarily mean it is supported by the platform.

To determine which MIBs are included in other releases, see the “Downloading and Compiling MIBs” section.

Platform-Independent MIB Categories

The MIBs are categorized into three types:

- see the “Supported and Verified MIBs” section
- see the “Supported and Unverified MIBs” section
- see the “Unverified or Unsupported MIBs” section

Supported and Verified MIBs

The MIB exists in the image, the code is implemented, and Cisco has verified that all the supported objects work properly. These MIBs are tested for the IOS XR platforms.

Supported and Unverified MIBs

The MIB exists in the image, the code is implemented, but Cisco has not verified if it is working properly. In other words, the user may get something if they query the MIB. However, the information may be correct or incorrect if the MIB has not been tested. These MIBs are not tested for the platform support.

Unverified or Unsupported MIBs

The MIB exists in the image but is either not tested or not supported. These MIBs are neither tested nor supported for the IOS XR platforms.

Note The CISCO-IPSEC-FLOW-MONITOR-MIB, CISCO-IPSEC-MIB, and CISCO-IPSEC-POLICY-MAP-MIB are not supported in Release 4.2.

MIB Support Category Summary

IGP/EGP Routing Protocol MIB Support Summary

This table is a summary of IOS-XR IGP/EGP Routing MIB Support.

Table 1-1 MIB Support Summary: IOS-XR Routing

Area	MIBs	Description	IPv6 Support	3.8/3.9 Mods
OSPF	OSPF-MIB, OSPF-TRAP-MIB, OSPFv3-MIB	OSPF Management	Yes (OSPFv3-MIB)	Initialize NSSA translators (ospfv3AreaNssaTranslatorState, ospfv3AreaNssaTranslatorRole) [CSCtb27

Multicast MIB Support Summary

This table is a summary of IOS-XR Multicast MIB Support.

MSD$

Table 1-2 MIB Support Summary: IOS-XR Multicast

Area	MIBs	Description	IPv6 Support	3.8/3.9 Mods	4.0 Mods/Roadmap
General (Multicast protocol independent)	CISCO-IPMROUTE-MIB	Management of IP Multicast routing in a manner independent of the specific multicast routing protocol in use (multicast packet/octet counters, and so forth.)	Yes	None	VRF support added in 4.0/no additional planned improvements

Table 1-2 MIB Support Summary: IOS-XR Multicast

Area	MIBs	Description	IPv6 Support	3.8/3.9 Mods	4.0 Mods/Roadmap
PIM	CISCO-PIM-MIB (traps only), CISCO-IETF-PIM-MIB,CISCO-IETF-PIM-EXT-MIB	PIM management	Yes	None	VRF support added in 4.0/no additional planned improvements
IGMP/MLD	MGMD-STD-MIB (CISCOized version of RFC 2933, IGMP-STD MIB), IPV6-MLD-MIB	IGMP (Multicast group management)/MLD (Multicast Listener Discovery) management	Yes	None	No planned improvements (MLD not VRFized)
MSDP	CISCO-IETF-MS DP-MIB	MSDP[1] management	v4 only feature, no v6 support	None	No planned improvements (MLD not VRFized)
Multicast VPN	None	Multicast VPN management	—	—	CISCO-MVPN MIB under consideration for 4.1

[1] MSDP = multicast source discovery protocol

L2VPN MIB Support Summary

This table is a summary of IOS-XR L2VPN MIB Support.

Table 1-3 MIB Support Summary: IOS-XR L2VPN

Area	MIBs	Description	IPv6 Support	3.8/3.9 Mods	4.0 Mods/Road-map
General PW support	PW-MIB	Cisco version of standard	—	None	No additional planned improvements
PW over Ethernet transport	PW-ENET	Cisco version of standard	—	None	No additional planned improvements
PW MPLS	PW-MPLS	Cisco version of standard	—	None	No additional Planned improvements
Textual Conventions	PW-TC	Cisco version of standard	—	None	No additional planned improvements

IPv6 MIB Support Summary

This table is a summary of IOS-XR IPv6 MIB Support.

Table 1-4 MIB Support Summary: IOS-XR IPv6

Area	MIBs	IPv6 Support	IPv6 Support Details
IP	IP-MIB, IP-FORWARD-MIB	Yes	Consistent with IOS (in most cases), IPV4 support provided by older RFC MIB. IPV6 support provided in newer MIB (IPV6-MIB, IPV6-TC-MIB).
OSPF	OPSFv3-MIB	Yes	Yes
Multicast	CISCO-IETF-PIM-MIB, CISCO-IETF-PIM-EXT-MIB, CISCO-IPMROUTE-MIB, MGMD-STD-MIB, IPV6-MLD-MIB	Yes	Additions to Cisco RFCs/MIBs to provide support for v6 addresses.
TCP, UDP	TCP-MIB, UDP-MIB	Yes	RFC standard IP version independent support.
BGP	BGP-MIB, CISCO-BGP-MIB	Yes	Cisco IPv6 support
Ping, ICMP	CISCO-PING-MIB	No	–
DHCP, Address Pool	–	–	–

MIB Version String Description

The MIB version string indicates the date and time that the module was most recently modified. The format is YYMMDDHHMMZ or YYYYMMDDHHMMZ, where:

- YY is the last two digits of the year (only years between 1900 and 1999).
- YYYY is all four digits of the year (any year).
- MM is the month (01 through 12).
- DD is the day of the month (01 through 31).
- HH is hours (00 through 23).
- MM is minutes (00 through 59).
- Z (the ASCII character Z) denotes Coordinated Universal Time (UTC, formerly Greenwich Mean Time, GMT).

This datatype stores the date and time fields YEAR, MONTH, DAY, HOUR, MINUTE, SECOND, TIMEZONE_HOUR, and TIMEZONE_MINUTE.

Note For example, 9502192015Z and 199502192015Z represent 8:15 GMT on 19 February 1995. Years after 1999 use the four-digit format. Years 1900–1999 may use the two or four digit format.

Note In the following table, the term *Revision not available* refers to the MIB module that does not have a recorded time stamp indicating the latest modification.

Platform Independent MIBs

This table lists the Platform-Independent MIBs.

Table 1-5 *Platform-Independent MIBs*

	midb process	Supported		Unsupported or Unverified	Not in Image
		Verified	Unverified		
ATM-MIB[1]	mibd-interface				
■ Release 3.7		9810191200Z			
■ Release 3.9		9810191200Z			
■ Release 4.0		9810191200Z			
■ Release 4.2		9810191200Z			
■ Release 4.3		9810191200Z			
ATM-FORUM-MIB[1]					
■ Release 3.7	–	Revision not available			
■ Release 3.9		Revision not available			
■ Release 4.0		Revision not available			
■ Release 4.2		Revision not available			
■ Release 4.3		Revision not available			
ATM2-MIB[1]	mibd-interface				
■ Release 3.7		200309230000Z			
■ Release 3.9		200309230000Z			
■ Release 4.0		200309230000Z			
■ Release 4.2		200309230000Z			
■ Release 4.3		200309230000Z			
BGP4-MIB	mibd-route				
■ Release 3.7		RFC 4273			
■ Release 3.9		RFC 4273			

Table 1-5 Platform-Independent MIBs

	midb process	Supported		Unsupported or Unverified	Not in Image
		Verified	Unverified		
■ Release 4.0		RFC 4273			
■ Release 4.2		RFC 4273			
■ Release 4.3		RFC 4273			
BRIDGE-MIB[1]	mibd-interface				
■ Release 3.7		RFC 4188			
■ Release 3.9		RFC 4188			
■ Release 4.0		RFC 4188			
■ Release 4.2		RFC 4188			
■ Release 4.3		RFC 4188			
CISCO-ATM-EXT-MIB[1]	mibd-interface				
■ Release 3.7		200301060000Z			
■ Release 3.9		200301060000Z			
■ Release 4.0		200301060000Z			
■ Release 4.2		200301060000Z			
■ Release 4.3		200301060000Z			
CISCO-ATM-QOS-MIB[1]	mibd-interface				
■ Release 3.7		200206100000Z			
■ Release 3.9		200206100000Z			
■ Release 4.0		200206100000Z			
■ Release 4.2		200206100000Z			
■ Release 4.3		200206100000Z			
CISCO-BGP4-MIB	mibd-route				
■ Release 3.7		200302240000Z			
■ Release 3.9		200302240000Z			
■ Release 4.0		200302240000Z			

Table 1-5 Platform-Independent MIBs

	midb process	Supported		Unsupported or Unverified	Not in Image
		Verified	Unverified		
■ Release 4.2		200302240000Z			
■ Release 4.3		200302240000Z			
CISCO-BGP-POLICY-ACCOUNTING-MIB	mibd-interface				
■ Release 3.7		200207260000Z			
■ Release 3.9		200207260000Z			
■ Release 4.0		200207260000Z			
■ Release 4.2		200207260000Z			
■ Release 4.3		200207260000Z			
CISCO-BULK-FILE-MIB[1]	mibd-infra				
■ Release 3.7		200206100000Z			
■ Release 3.9		200206100000Z			
■ Release 4.0		200206100000Z			
■ Release 4.2		200206100000Z			
■ Release 4.3		200206100000Z			
CISCO-CDP-MIB	mibd-interface				
■ Release 3.7		9812100000Z			
■ Release 3.9		9812100000Z			
■ Release 4.0		9812100000Z			
■ Release 4.2		9812100000Z			
■ Release 4.3		9812100000Z			
CISCO-CLASS-BASED-QOS-MIB1	mibd-interface				
■ Release 3.7		200901260000Z			
■ Release 3.9		200901260000Z			

Table 1-5 Platform-Independent MIBs

	midb process	Supported		Unsupported or Unverified	Not in Image
		Verified	Unverified		
■ Release 4.0		200901260000Z			
■ Release 4.2		200901260000Z			
■ Release 4.3		200901260000Z			
CISCO-CONFIG-COPY-MIB	mibd-infra				
■ Release 3.7		200504060000Z			
■ Release 3.9		200504060000Z			
■ Release 4.0		200504060000Z			
■ Release 4.2		200504060000Z			
■ Release 4.3		200504060000Z			
CISCO-CONFIG-MAN-MIB	mibd-infra				
■ Release 3.7		200704270000Z			
■ Release 3.9		200704270000Z			
■ Release 4.0		200704270000Z			
■ Release 4.2		200704270000Z			
■ Release 4.3		200704270000Z			
CISCO-CONTEXT-MAPPING-MIB	mibd-infra				
■ Release 3.7		200811220000Z			
■ Release 3.9		200811220000Z			
■ Release 4.0		200811220000Z			
■ Release 4.2		200811220000Z			
■ Release 4.3		200811220000Z			
CISCO-DS3-MIB[1]	mibd-interface				
■ Release 3.7		200205210000Z			
■ Release 3.9		200205210000Z			
■ Release 4.0		200205210000Z			

Table 1-5 Platform-Independent MIBs

	midb process	Supported		Unsupported or Unverified	Not in Image
		Verified	Unverified		
■ Release 4.2		200205210000Z			
■ Release 4.3		200205210000Z			
CISCO-ENHANCED-IMAGE- MIB[1]	mibd-entity				
■ Release 3.7		200501060000Z			
■ Release 3.9		200501060000Z			
■ Release 4.0		200501060000Z			
■ Release 4.2		200501060000Z			
■ Release 4.3		200501060000Z			
CISCO-ENHANCED-MEMPOOL-MIB[1]	mibd-entity				
■ Release 3.7		200812050000Z			
■ Release 3.9		200812050000Z			
■ Release 4.0		200812050000Z			
■ Release 4.2		200812050000Z			
■ Release 4.3		200812050000Z			
CISCO-ENTITY-ASSET-MIB[1]	mibd-entity				
■ Release 3.7		200309180000Z			
■ Release 3.9		200309180000Z			
■ Release 4.0		200309180000Z			
■ Release 4.2		200309180000Z			
■ Release 4.3		200309180000Z			
CISCO-ENTITY-FRU-CONTROL-MIB[1]	mibd-entity				

Table 1-5 Platform-Independent MIBs

	midb process	Supported		Unsupported or Unverified	Not in Image
		Verified	Unverified		
■ Release 3.7		200810080000Z			
■ Release 3.9		200810080000Z			
■ Release 4.0		200810080000Z			
■ Release 4.2		200810080000Z			
■ Release 4.3		200810080000Z			
CISCO-ENTITY-REDUN-DANCY-MIB[1]	mibd-entity				
■ Release 3.7					
■ Release 3.9					
■ Release 4.0					
■ Release 4.2					
■ Release 4.3		200510010000Z			
CISCO-ENTITY-SEN-SOR-MIB[1]	mibd-entity				
■ Release 3.7		200711120000Z			
■ Release 3.9		200711120000Z			
■ Release 4.0		200711120000Z			
■ Release 4.2		200711120000Z			
■ Release 4.3		200711120000Z			
CISCO-ENTITY-STATE-EXT- MIB[1]	mibd-infra				
■ Release 3.7		201006160000Z			
■ Release 3.9		201006160000Z			
■ Release 4.0		201006160000Z			
■ Release 4.2		201006160000Z			

Table 1-5 Platform-Independent MIBs

	midb process	Supported		Unsupported or Unverified	Not in Image
		Verified	Unverified		
■ Release 4.3		201006160000Z			
CISCO-FABRIC-HFR-MIB[1]	mibd-infra				
■ Release 3.7		200904140000Z			
■ Release 3.9		200904140000Z			
■ Release 4.0		200904140000Z			
■ Release 4.2		200904140000Z			
■ Release 4.3		200904140000Z			
CISCO-FLASH-MIB[1]	mibd-infra				
■ Release 3.7		200906030000Z			
■ Release 3.9		200906030000Z			
■ Release 4.0		200906030000Z			
■ Release 4.2		200906030000Z			
■ Release 4.3		200906030000Z			
CISCO-FRAME-RELAY-MIB[1]	mibd-interface				
■ Release 3.7		200010130000Z			
■ Release 3.9		200010130000Z			
■ Release 4.0		200010130000Z			
■ Release 4.2		200010130000Z			
CISCO-FTP-CLIENT-MIB[1]	mibd-infra				
■ Release 3.7		200603310000Z			
■ Release 3.9		200603310000Z			
■ Release 4.0		200603310000Z			
■ Release 4.2		200603310000Z			
CISCO-HSRP-EXT-MIB	mibd-interface				

Table 1-5 Platform-Independent MIBs

	midb process	Supported		Unsupported or Unverified	Not in Image
		Verified	**Unverified**		
■ Release 3.7		9808030000Z			
■ Release 3.9		9808030000Z			
■ Release 4.0		9808030000Z			
■ Release 4.2		9808030000Z			
■ Release 4.3		9808030000Z			
CISCO-HSRP-MIB	mibd-interface				
■ Release 3.7		9808030000Z			
■ Release 3.9		9808030000Z			
■ Release 4.0		9808030000Z			
■ Release 4.2		9808030000Z			
■ Release 4.3		9808030000Z			
CISCO-IETF-BFD-MIB	mibd-route				
■ Release 3.7		200804240000Z			
■ Release 3.9		200804240000Z			
■ Release 4.0		200804240000Z			
■ Release 4.2		200804240000Z			
■ Release 4.3		200804240000Z			
CISCO-IETF-FRR-MIB	Mibd route				
■ Release 3.7		200804291200Z			
■ Release 3.9		200804291200Z			
■ Release 4.0		200804291200Z			
■ Release 4.2		200804291200Z			
■ Release 4.3		200804291200Z			
CISCO-IETF-IPMROUTE-MIB	mibd-interface				
■ Release 3.7		200608240000Z			

Table 1-5 Platform-Independent MIBs

	midb process	**Supported**		**Unsupported or Unverified**	**Not in Image**
		Verified	**Unverified**		
▪ Release 3.9		200608240000Z			
▪ Release 4.0		200608240000Z			
▪ Release 4.2		200608240000Z			
▪ Release 4.3		200608240000Z			
CISCO-IETF-MSDP-MIB	mibd-interface				
▪ Release 3.7		200605190000Z			
▪ Release 3.9		200605190000Z			
▪ Release 4.0		200605190000Z			
▪ Release 4.2		200605190000Z			
▪ Release 4.3		200605190000Z			
CISCO-IETF-PIM-MIB	–				
▪ Release 3.7		200502220000Z			
▪ Release 3.9		200502220000Z			
▪ Release 4.0		200502220000Z			
▪ Release 4.2		200502220000Z			
▪ Release 4.3		200502220000Z			
CISCO-IETF-PIM-EXT-MIB	mibd-interface				
▪ Release 3.7		200608250000Z			
▪ Release 3.9		200608250000Z			
▪ Release 4.0		200608250000Z			
▪ Release 4.2		200608250000Z			
▪ Release 4.3		200608250000Z			
CISCO-IETF-PW-MIB	mibd-interface				
▪ Release 3.7		200512200000Z			

Table 1-5 Platform-Independent MIBs

	midb process	Supported		Unsupported or Unverified	Not in Image
		Verified	**Unverified**		
- Release 3.9		200512200000Z			
- Release 4.0		200512200000Z			
- Release 4.2		200512200000Z			
- Release 4.3		200512200000Z			
CISCO-IETF-PW-ENET-MIB	mibd-interface				
- Release 3.7		200209221200Z			
- Release 3.9		200209221200Z			
- Release 4.0		200209221200Z			
- Release 4.2		200209221200Z			
- Release 4.3		200209221200Z			
CISCO-IETF-PW-MPLS-MIB	mibd-interface				
- Release 3.7		200302261200Z			
- Release 3.9		200302261200Z			
- Release 4.0		200302261200Z			
- Release 4.2		200302261200Z			
- Release 4.3		200302261200Z			
CISCO-IETF-VPLS-BGP-EXT-MIB	mibd-interface				
- Release 3.7		200810240000Z			
- Release 3.9		200810240000Z			
- Release 4.0		200810240000Z			
- Release 4.2		200810240000Z			
- Release 4.3		200810240000Z			

Table 1-5 Platform-Independent MIBs

	midb process	**Supported**		**Unsupported or Unverified**	**Not in Image**
		Verified	**Unverified**		
CISCO-IETF-VPLS-GENERIC- MIB[1]	mibd-interface				
■ Release 3.7		200710221200Z			
■ Release 3.9		200710221200Z			
■ Release 4.0		200710221200Z			
■ Release 4.2		200710221200Z			
■ Release 4.3		200710221200Z			
CISCO-IETF-VPLS-LDP-MIB[1]	mibd-interface				
■ Release 3.7		200711221200Z			
■ Release 3.9		200711221200Z			
■ Release 4.0		200711221200Z			
■ Release 4.2		200711221200Z			
■ Release 4.3		200711221200Z			
CISCO-IF-EXTENSION-MIB[1]	mibd-interface				
■ Release 3.7		200707230000Z			
■ Release 3.9					
■ Release 4.0					
■ Release 4.2					
■ Release 4.3					
CISCO-IP-STAT-MIB	mibd-interface				
■ Release 3.7		200112202300Z			
■ Release 3.9		200112202300Z			
■ Release 4.0		200112202300Z			

Table 1-5 Platform-Independent MIBs

	midb process	**Supported**		**Unsupported or Unverified**	**Not in Image**
		Verified	**Unverified**		
■ Release 4.2 ■ Release 4.3		200112202300Z 200112202300Z			
CISCO-MEMORY-POOL-MIB[1] ■ Release 3.7 ■ Release 3.9 ■ Release 4.0 ■ Release 4.2 ■ Release 4.3	mibd-infra	200107310000Z 200107310000Z 200107310000Z 200107310000Z 200107310000Z			
CISCO-NTP-MIB ■ Release 3.7 ■ Release 3.9 ■ Release 4.0 ■ Release 4.2 ■ Release 4.3	mibd-interface	200607310000Z 200607310000Z 200607310000Z 200607310000Z 200607310000Z			
CISCO-OTN-IF-MIB ■ Release 3.7 ■ Release 3.9 ■ Release 4.0 ■ Release 4.2 ■ Release 4.3	mibd-interface	200710100000Z 200710100000Z 200710100000Z 200710100000Z 200710100000Z			
CISCO-PIM-MIB ■ Release 3.7 ■ Release 3.9	mibd-interface	200011020000Z 200011020000Z			

Table 1-5 Platform-Independent MIBs

	midb process	Supported		Unsupported or Unverified	Not in Image
		Verified	**Unverified**		
■ Release 4.0		200011020000Z			
■ Release 4.2		200011020000Z			
■ Release 4.3		200011020000Z			
CISCO-PING-MIB	mibd-route				
■ Release 3.7		200108280000Z			
■ Release 3.9		200108280000Z			
■ Release 4.0		200108280000Z			
■ Release 4.2		200108280000Z			
■ Release 4.3		200108280000Z			
CISCO-PROCESS-MIB[1]	mibd-entity				
■ Release 3.7		200910120000Z			
■ Release 3.9		200910120000Z			
■ Release 4.0		200910120000Z			
■ Release 4.2		200910120000Z			
■ Release 4.3		200910120000Z			
CISCO-RF-MIB[1]	mibd-infra				
■ Release 3.7		200803180000Z			
■ Release 3.9		200803180000Z			
■ Release 4.0		200803180000Z			
■ Release 4.2		200803180000Z			
■ Release 4.3		200803180000Z			
CISCO-RTTMON-MIB	mibd-infra				
■ Release 3.7		200803240000Z			
■ Release 3.9		200803240000Z			
■ Release 4.0		200803240000Z			

Table 1-5 *Platform-Independent MIBs*

	midb process	Supported		Unsupported or Unverified	Not in Image
		Verified	Unverified		
■ Release 4.2		200803240000Z			
■ Release 4.3		200803240000Z			
CISCO-SONET-MIB[1]	mibd-interface				
■ Release 3.7		200303070000Z			
■ Release 3.9		200303070000Z			
■ Release 4.0		200303070000Z			
■ Release 4.2		200303070000Z			
■ Release 4.3		200303070000Z			
CISCO-SYSLOG-MIB	mibd-infra				
■ Release 3.7		200512030000Z			
■ Release 3.9		200512030000Z			
■ Release 4.0		200512030000Z			
■ Release 4.2		200512030000Z			
■ Release 4.3		200512030000Z			
CISCO-SYSTEM-MIB	mibd-infra				
■ Release 3.7		200709160000Z			
■ Release 3.9		200709160000Z			
■ Release 4.0		200709160000Z			
■ Release 4.2		200709160000Z			
■ Release 4.3		200709160000Z			
CISCO-TCP-MIB	mibd-route				
■ Release 3.7		200111120000Z			
■ Release 3.9		200111120000Z			
■ Release 4.0		200111120000Z			
■ Release 4.2		200111120000Z			

Table 1-5 Platform-Independent MIBs

	midb process	Supported		Unsupported or Unverified	Not in Image
		Verified	**Unverified**		
■ Release 4.3		200111120000Z			
CISCO-VLAN-IFTABLE-RELATIONSHIP-MIB	mibd-interface				
■ Release 3.7		9904010530Z			
■ Release 3.9		9904010530Z			
■ Release 4.0		9904010530Z			
■ Release 4.2		9904010530Z			
■ Release 4.3		9904010530Z			
DS1-MIB[1]	mibd-interface				
■ Release 3.7		9808011830Z			
■ Release 3.9		9808011830Z			
■ Release 4.0		9808011830Z			
■ Release 4.2		9808011830Z			
■ Release 4.3		9808011830Z			
DS3-MIB[1]	mibd-interface				
■ Release 3.7		200205210000Z			
■ Release 3.9		200205210000Z			
■ Release 4.0		200205210000Z			
■ Release 4.2		200205210000Z			
■ Release 4.3		200205210000Z			
ENTITY-MIB (RFC 2737)[1]	mibd-entity				
■ Release 3.7		RFC 2737			
■ Release 3.9		RFC 2737			
■ Release 4.0		RFC 2737			

Table 1-5 Platform-Independent MIBs

	midb process	Supported		Unsupported or Unverified	Not in Image
		Verified	Unverified		
■ Release 4.2		RFC 2737			
■ Release 4.3		RFC 2737			
ENTITY-STATE-MIB	mibd-entity				
■ Release 3.7		200511220000Z			
■ Release 3.9		200511220000Z			
■ Release 4.0		200511220000Z			
■ Release 4.2		200511220000Z			
■ Release 4.3		200511220000Z			
EVENT-MIB	mibd-infra				
■ Release 3.7		RFC 2981			
■ Release 3.9		RFC 2981			
■ Release 4.0		RFC 2981			
■ Release 4.2		RFC 2981			
■ Release 4.3		RFC 2981			
EXPRESSION-MIB	mibd-infra				
■ Release 3.7		200511240000Z			
■ Release 3.9		200511240000Z			
■ Release 4.0		200511240000Z			
■ Release 4.2		200511240000Z			
■ Release 4.3		200511240000Z			
FRAME-RELAY-DTE-MIB[1]	mibd-interface				
■ Release 3.7		9705010229Z			
■ Release 3.9		9705010229Z			
■ Release 4.0		9705010229Z			
■ Release 4.2		9705010229Z			
■ Release 4.3		9705010229Z			

Table 1-5 Platform-Independent MIBs

	midb process	Supported		Unsupported or Unverified	Not in Image
		Verified	Unverified		
IEEE8023-LAG-MIB	mibd-interface				
■ Release 3.7		200006270000Z			
■ Release 3.9		200006270000Z			
■ Release 4.0		200006270000Z			
■ Release 4.2		200006270000Z			
■ Release 4.3		200006270000Z			
IF-MIB (RFC 2863)[1]	mibd-interface				
■ Release 3.7		RFC 2233			
■ Release 3.9		RFC 2233			
■ Release 4.0		RFC 2233			
■ Release 4.2		RFC 2233			
■ Release 4.3		RFC 2233			
IP-FORWARD-MIB	mibd-route				
■ Release 3.7		RFC 2096 RFC 4292			
■ Release 3.9		RFC 2096 RFC 4292			
■ Release 4.0		RFC 2096 RFC 4292			
■ Release 4.2		RFC 2096 RFC 4292			
■ Release 4.3		RFC 2096 RFC 4292			
IP-MIB	mibd-interface				
■ Release 3.7		RFC 2011			

Table 1-5 Platform-Independent MIBs

	midb process	**Supported**		**Unsupported or Unverified**	**Not in Image**
		Verified	**Unverified**		
■ Release 3.9		RFC 2011			
■ Release 4.0		RFC 2011			
■ Release 4.2		RFC 2011			
■ Release 4.3		RFC 2011			
IPV6-MIB	mibd-interface				
■ Release 3.7		9802052155Z			
■ Release 3.9		9802052155Z			
■ Release 4.0		9802052155Z			
■ Release 4.2		9802052155Z			
■ Release 4.3		9802052155Z			
IPV6-MLD-MIB	mibd-interface				
■ Release 3.7		200101250000Z			
■ Release 3.9		200101250000Z			
■ Release 4.0		200101250000Z			
■ Release 4.2		200101250000Z			
■ Release 4.3		200101250000Z			
ISIS-MIB	mibd-route				
■ Release 3.7		200604040000Z			
■ Release 3.9		200604040000Z			
■ Release 4.0		200604040000Z			
■ Release 4.2		200604040000Z			
■ Release 4.3		200604040000Z			
MPLS-L3VPN-STD-MIB	mibd-route				
■ Release 3.7		200601230000Z			
■ Release 3.9		200601230000Z			
■ Release 4.0		200601230000Z			

Table 1-5 Platform-Independent MIBs

	midb process	**Supported**		**Unsupported or Unverified**	**Not in Image**
		Verified	**Unverified**		
■ Release 4.2		200601230000Z			
■ Release 4.3		200601230000Z			
MPLS-LDP-GENERIC-STD-MIB	mibd-route				
■ Release 3.7		200406030000Z			
■ Release 3.9		200406030000Z			
■ Release 4.0		200406030000Z			
■ Release 4.2		200406030000Z			
■ Release 4.3		200406030000Z			
MPLS-LDP-GENERIC-STD-MIB	mibd-route				
■ Release 3.7		200406030000Z			
■ Release 3.9		200406030000Z			
■ Release 4.0		200406030000Z			
■ Release 4.2		200406030000Z			
■ Release 4.3		200406030000Z			
MPLS-LDP-STD-MIB	mibd-route				
■ Release 3.7		200406030000Z			
■ Release 3.9		200406030000Z			
■ Release 4.0		200406030000Z			
■ Release 4.2		200406030000Z			
■ Release 4.3		200406030000Z			
MPLS-LSR-STD-MIB	mibd-route				
■ Release 3.7		200406030000Z			
■ Release 3.9		200406030000Z			
■ Release 4.0		200406030000Z			

Table 1-5 Platform-Independent MIBs

	midb process	Supported		Unsupported or Unverified	Not in Image
		Verified	**Unverified**		
■ Release 4.2		200406030000Z			
■ Release 4.3		200406030000Z			
MPLS-TE-STD-MIB	mibd-route				
■ Release 3.7		RFC 3812			
■ Release 3.9		RFC 3812			
■ Release 4.0		RFC 3812			
■ Release 4.2		RFC 3812			
■ Release 4.3		RFC 3812			
NOTIFICATION-LOG-MIB	mibd-infra				
■ Release 3.7		200011270000Z			
■ Release 3.9		200011270000Z			
■ Release 4.0		200011270000Z			
■ Release 4.2		200011270000Z			
■ Release 4.3		200011270000Z			
OSPF-MIB	mibd-route				
■ Release 3.7					
■ Release 3.9		200611100000Z			
■ Release 4.0		200611100000Z			
■ Release 4.2		200611100000Z			
■ Release 4.3		200611100000Z			
OSPF-TRAP-MIB	—	200611100000Z			
■ Release 3.7					
■ Release 3.9		200611100000Z			
■ Release 4.0		200611100000Z			
■ Release 4.2		200611100000Z			
■ Release 4.3		200611100000Z			

Table 1-5 Platform-Independent MIBs

	midb process	Supported		Unsupported or Unverified	Not in Image
		Verified	Unverified		
OSPFV3-MIB	mibd-route	200611100000Z			
■ Release 3.7		200709171200Z			
■ Release 3.9		200709171200Z			
■ Release 4.0		200709171200Z			
■ Release 4.2		200709171200Z			
■ Release 4.3		200709171200Z			
RADIUS-ACC-CLIENT-MIB[1]	mibd-infra				
■ Release 3.7		200300000000Z			
■ Release 3.9		200300000000Z			
■ Release 4.0		200300000000Z			
■ Release 4.2		200300000000Z			
■ Release 4.3		200300000000Z			
RADIUS-AUTH-CLIENT-MIB[1]	mibd-infra				
■ Release 3.7		200300000000Z			
■ Release 3.9		200300000000Z			
■ Release 4.0		200300000000Z			
■ Release 4.2		200300000000Z			
■ Release 4.3		200300000000Z			
RFC 1213-MIB[1]	–				
■ Release 3.7		Revision not available			
■ Release 3.9		Revision not available			
■ Release 4.0		Revision not available			
■ Release 4.2		Revision not available			
■ Release 4.3		Revision not available			
RSVP-MIB[1]	mibd-				

Table 1-5 Platform-Independent MIBs

	midb process	**Supported**		**Unsupported or Unverified**	**Not in Image**
		Verified	**Unverified**		
	interface				
■ Release 3.7		9808251820Z			
■ Release 3.9		9808251820Z			
■ Release 4.0		9808251820Z			
■ Release 4.2		9808251820Z			
■ Release 4.3		9808251820Z			
SNMP-COMMUNITY-MIB (RFC 2576)	snmpd				
■ Release 3.7		200210140000Z			
■ Release 3.9		200210140000Z			
■ Release 4.0		200210140000Z			
■ Release 4.2		200210140000Z			
■ Release 4.3		200210140000Z			
SNMP-FRAMEWORK -MIB (RFC 2571)	snmpd				
■ Release 3.7		200210140000Z			
■ Release 3.9		200210140000Z			
■ Release 4.0		200210140000Z			
■ Release 4.2		200210140000Z			
■ Release 4.3		200210140000Z			
SNMP-MPD-MIB (RFC 2572)	snmpd				
■ Release 3.7		9905041636Z			
■ Release 3.9		9905041636Z			
■ Release 4.0		9905041636Z			
■ Release 4.2		9905041636Z			
■ Release 4.3		9905041636Z			

Table 1-5 Platform-Independent MIBs

	midb process	Supported		Unsupported or Unverified	Not in Image
		Verified	Unverified		
SNMP-NOTIFICATION-MIB (RFC 2573)	snmpd				
■ Release 3.7		9808040000Z			
■ Release 3.9		9808040000Z			
■ Release 4.0		9808040000Z			
■ Release 4.2		9808040000Z			
■ Release 4.3		9808040000Z			
SNMP-TARGET-MIB (RFC	snmpd				
■ Release 3.7		9808040000Z			
■ Release 3.9		9808040000Z			
■ Release 4.0		9808040000Z			
■ Release 4.2		9808040000Z			
■ Release 4.3		9808040000Z			
SNMP-USM-MIB (RFC 2574)	snmpd				
■ Release 3.7		9901200000Z			
■ Release 3.9		9901200000Z			
■ Release 4.0		9901200000Z			
■ Release 4.2		9901200000Z			
■ Release 4.3		9901200000Z			
SNMP-VACM-MIB (RFC	snmpd				
■ Release 3.7		RFC 2575			
■ Release 3.9		RFC 2575			
■ Release 4.0		RFC 2575			
■ Release 4.2		RFC 2575			
■ Release 4.3		RFC 2575			
SNMPv2-MIB (RFC	snmpd				

Table 1-5 Platform-Independent MIBs

	midb process	Supported		Unsupported or Unverified	Not in Image
		Verified	Unverified		
1907)					
■ Release 3.7		RFC 1904			
■ Release 3.9		RFC 1904			
■ Release 4.0		RFC 1904			
■ Release 4.2		RFC 1904			
■ Release 4.3		RFC 1904			
SONET-MIB[1]	mibd-interface				
■ Release 3.7		200308110000Z			
■ Release 3.9		200308110000Z			
■ Release 4.0		200308110000Z			
■ Release 4.2		200308110000Z			
■ Release 4.3		200308110000Z			
TCP-MIB	mibd-route				
■ Release 3.7		200502180000Z			
■ Release 3.9		200502180000Z			
■ Release 4.0		200502180000Z			
■ Release 4.2		200502180000Z			
■ Release 4.3		200502180000Z			
UDP-MIB	mibd-route				
■ Release 3.7		200505200000Z			
■ Release 3.9		200505200000Z			
■ Release 4.0		200505200000Z			
■ Release 4.2		200505200000Z			
■ Release 4.3		200505200000Z			
VRRP-MIB	–				
■ Release 3.7					Y
■ Release 3.9		200003030000Z			

Table 1-5 Platform-Independent MIBs

	midb process	Supported		Unsupported or Unverified	Not in Image
		Verified	Unverified		
■ Release 4.0		200003030000Z			
■ Release 4.2		200003030000Z			
■ Release 4.3		200003030000Z			

[1] These MIBs may have a different behavior on a per-platform basis.

TC-MIBs

This tabe lists the TC (Textual Conventions) MIBs. These MIBs are verified but cannot be queried.

Table 1-6 TC MIBs

MIB	Verified
CISCO-IETF-PW-TC-MIB	200607211200Z
IPV6-TC	Revision not available
MPLS-TC-STD-MIB	200406030000Z
VPN-TC-STD-MIB	200511150000Z

MIB Notification Names of the Platform-Independent MIBs

This table lists the Notification Names associated with platform-independent MIBs.

Table 1-7 MIB Notification Names in the Platform-independent MIBs

MIB	Notification Name
BGP4-MIB	bgpEstablishedNotification bgpBackwardTransNotiifcation
BRIDGE-MIB	newRoot, topologyChange
CISCO-BGP4-MIB	cbgpFsmStateChange, cbgpBackwardTransition, cbgpPrefixThresholdExceeded, cbgpPrefixThresholdClear
CISCO-BULK-FILE-MIB	cbfDefineFileCompletion
CISCO-CONFIG-COPY-MIB	ccCopyCompletion
CISCO-CONFIG-MAN-MIB	ciscoConfigManEvent

Table 1-7 MIB Notification Names in the Platform-independent MIBs

MIB	Notification Name
CISCO-ENTITY-FRU-CONTROL-MIB	cefcModuleStatusChange, cefcPowerStatusChange, cefcFRUInserted, cefcFRURemoved, cefcFanTrayStatusChange
CISCO-ENTITY-SENSOR-MIB	ciscoFlashCopyCompletionTrap, ciscoFlashDeviceInsertedNotif, ciscoFlashDeviceRemovedNotif, ciscoFlashMiscOpCompletionTrap
CISCO-HSRP-MIB	cHsrpStateChange
CISCO-IETF-PIM-MIB	cPimNbrLoss
CISCO-IETF-PIM-EXT-MIB	ciscoIetfPimExtInterfaceUp
	ciscoIetfPimExtInterfaceDown
	ciscoIetfPimExtRPMappingChange
CISCO-IETF-PW-MIB	cpwVcDown
	cpwVcUp
CISCO-NTP-MIB	cntpHighPriorityConnFailure
	cntpHighPriorityConnRestore
	ciscoNtpSrvStatusChange
	cntpGeneralConnFailure
	cntpHighPriorityConnRestore
CISCO-PIM-MIB	ciscoPimInvalidRegister
	ciscoPimInvalidJoinPrune
CISCO-RF-MIB	ciscoRFSwactNotif,
	ciscoRFProgressionNotif
CISCO-SYSLOG-MIB	clogMessageGenerated
ENTITY-MIB (RFC 2737)	mteTriggerFired,
	mteTriggerRising,

Table 1-7 MIB Notification Names in the Platform-independent MIBs

MIB	Notification Name
	mteTriggerFalling,
	mteTriggerFailure,
	mteEventSetFailure
ISIS-MIB	isisDatabaseOverload
	isisCorruptedLSPDetected
	isisAttemptToExceedMaxSequence
	isisIDLenMismatch
	isisMaxAreaAddressesMismatch
	isisOwnLSPPurge
	isisSequenceNumberSkip
	isisAuthenticationTypeFailure
	isisAuthenticationFailure
	isisVersionSkew
	isisAreaMismatch
	isisRejectedAdjacency
	isisLSPTooLargeToPropagate
	isisOrigLSPBuffSizeMismatch
	isisProtocolsSupportedMismatch
	isisAdjacencyChange
	isisLSPErrorDetected
MPLS-L3VPN-STD-MIB	mplsL3VpnVrfUp
	mplsL3VpnVrfDown
	mplsL3VpnRouteMidThreshExceeded
	mplsL3VpnVrfNumVrfRouteMaxThreshExceeded
	mplsL3VpnNumVrfRouteMaxThreshCleared

Table 1-7 MIB Notification Names in the Platform-independent MIBs

MIB	Notification Name
OSPFV3-MIB	ospfVirtIfStateChange
	ospfv3NbrStateChange
	ospfv3VirtNbrStateChange
	ospfv3IfConfigError
	ospfv3VirtIfConfigError
	ospfIfRxBadPacket
	ospfv3VirtIfRxBadPacket
	ospfv3LsdbOverflow
	ospfv3LsdbApproachingOverflow
	ospfv3IfStateChange
	ospfv3NssaTranslatorStatusChange
	ospfv3RestartStatusChange
	ospfv3NbrRestartHelperStatusChange
	ospfv3VirtNbrRestartHelperStatusChange
RFC 1213-MIB	newFlow
	lostFlow
VRRP-MIB	vrrpTrapNewMaster
	vrrpTrapAuthFailure

ATM-MIB

The ATM-MIB describes ATM and AAL5-related objects for managing ATM interfaces, ATM virtual links, ATM cross-connects, AAL5 entities, and AAL5 connections.

This table lists the tables associated with this MIB.

Table 1-8 ATM-MIB Tables and Descriptions

Name	Description
atmInterfaceConfTable	This table contains ATM local interface configuration parameters, one entry per ATM interface port.
atmInterfaceDs3PlcpTable	This table contains ATM interface DS3 PLCP parameters and state variables,

Table 1-8 ATM-MIB Tables and Descriptions

Name	**Description**
	one entry per ATM interface port.
atmInterfaceTCTable	This table contains ATM interface TC Sublayer parameters and state variables, one entry per ATM interface port.
atmTrafficDescrParamTable	This table contains information on ATM traffic descriptor type and the associated parameters.
atmVplTable	VPL table. A bi directional VPL is modeled as one entry in this table. This table can be used for PVCs, SVCs, and Soft PVCs. Entries are not present in this table for the VPIs used by entries in the atmVclTable.
atmVclTable	VCL table. A bi directional VCL is modeled as one entry in this table. This table can be used for PVCs, SVCs, and Soft PVCs.
atmVpCrossConnectTable	ATM VP Cross Connect table for PVCs. An entry in this table models two cross-connected VPLs. Each VPL must have its atmConnKind set to pvc(1).
atmVcCrossConnectTable	ATM VC Cross Connect table for PVCs. An entry in this table models two cross-connected VCLs. Each VCL must have its atmConnKind set to pvc(1).
aal5VccTable	This table contains AAL5 VCC performance parameters.

ATM-FORUM-MIB

The ATM-FORUM-MIB is one of the ATM Forum's ILMI MIBs, supporting the UNI 4.0 specification.

This table lists the tables associated with this MIB:

Table 1-9 ATM-FORUM-MIB Tables and Descriptions

Name	**Description**
atmfPortTable	Table of physical layer status and parameter information for the physical interface of ATM Interface.
atmfAtmLayerTable	Table of ATM layer status and parameter information for the ATM Interface.
atmfAtmStatsTable	This group is deprecated and should not be implemented except as required for backward compatibility with version 3.1 of the UNI specification.
atmfVpcTable	Table of status and parameter information on the virtual path connections which cross this ATM Interface. There is one entry in this table for each permanent virtual path connection.
atmfVpcAbrTable	Table of operational parameters related to the ABR virtual path connections which cross this ATM Interface. There is one entry in this table for each ABR virtual path connection. Each virtual path connection represented in this table must also be represented by an entry in the atmfVpcTable.
atmfVccTable	Table of status and parameter information on the virtual channel connections which are visible at this ATM Interface. There is one entry in this table for each permanent virtual channel connection, including reserved VCCs that are supported; that is, signaling, OAM flows, and ILMI, but not unassigned cells.
atmfVccAbrTable	Table of operational parameters related to the ABR virtual channel connections which cross this ATM Interface. There is one entry in this table for each ABR virtual channel connection. Each virtual channel connection represented in this table must also be represented by an entry in the atmfVccTable.

ATM2-MIB

The ATM2-MIB supplements the ATM-MIB as defined in RFC 2515.

This table lists the tables associated with this MIB.

Table 1-10 ATM2-MIB Tables and Descriptions

Name	**Description**
atmSvcVpCrossConnectTable	ATM SVPC Cross-Connect table. A bi directional VP cross-connect between two switched VPLs is modeled as one entry in this table. A Soft PVPC cross-connect, between a soft permanent VPL and a switched VPL, is also modeled as one entry in this table.
atmSvcVcCrossConnectTable	ATM SVCC Cross-Connect table. A bi directional VC cross-connect between two switched VCLs is modeled as one entry in this table. A Soft PVCC cross-connect, between a soft permanent VCL and a switched VCL, is also modeled as one entry in this table.
atmSigStatTable	This table contains ATM interface signaling statistics, one entry per ATM signaling interface.
atmSigSupportTable	This table contains ATM local interface configuration parameters, one entry per ATM signaling interface.
atmSigDescrParamTable	Table contains signaling capabilities of VCLs except the Traffic Descriptor. Traffic descriptors are described in the atmTrafficDescrParamTable.
atmIfRegisteredAddrTable	This table contains a list of ATM addresses that can be used for calls to and from a given interface by a switch or service. The ATM addresses are either registered by the endsystem via ILMI or statically configured. This table does not expose PNNI reachability information. ILMI registered addresses cannot be deleted using this table. This table only applies to switches and network services.
atmVclAddrTable	This table provides a mapping between the atmVclTable and the ATM called party/calling party address. This table can be used to retrieve the calling party and called party ATM address pair for a given VCL. Note that there can be more than one pair of calling party and called party ATM addresses for a VCL in a point to multi-point call.
atmAddrVclTable	This table provides an alternative way to retrieve the atmVclTable. This table can be used to retrieve the indexing to the atmVclTable by an ATM address.
atmVplStatTable	This table contains all statistics counters per VPL. It is used to monitor the usage of the VPL in terms of incoming cells and outgoing cells.
atmVplLogicalPortTable	Indicates whether the VPL is an ATM Logical Port interface (ifType = 80).
atmVclStatTable	This table contains all statistics counters per VCL. It is used to monitor the usage of the VCL in terms of incoming cells and outgoing cells.
atmAal5VclStatTable	This table provides a collection of objects providing AAL5 configuration and performance statistics of a VCL.
atmVclGenTable	General Information for each VC.
atmInterfaceExtTable	This table contains ATM interface configuration and monitoring information not defined in the atmInterfaceConfTable from the ATM-MIB. This includes the type of connection setup procedures, ILMI information, and information on the VPI/VCI range.
atmIlmiSrvcRegTable	This table contains a list of all the ATM network services known by this device. The characteristics of these services are made available through the ILMI,

Table 1-10 ATM2-MIB Tables and Descriptions

Name	Description
	using the ILMI general-purpose service registry MIB. These services may be made available to all ATM interfaces (atmIlmiSrvcRegIndex = 0) or to some specific ATM interfaces only (atmIlmiSrvcRegIndex = ATM interface index).
atmIlmiNetworkPrefixTable	Table specifying per-interface network prefix(es) supplied by the network side of the UNI during ILMI address registration. When no network prefixes are specified for a particular interface, one or more network prefixes based on the switch address(es) may be used for ILMI address registration.
atmVpCrossConnectXTable	This table contains one row per VP Cross-Connect represented in the atmVpCrossConnectTable.
atmVcCrossConnectXTable	This table contains one row per VC Cross-Connect represented in the atmVcCrossConnectTable.
atmCurrentlyFailingPVplTable	Table indicating all VPLs for which there is an active row in the atmVplTable having an atmVplConnKind value of pvc and an atmVplOperStatus with a value other than up.
atmCurrentlyFailingPVclTable	Table indicating all VCLs for which there is an active row in the atmVclTable having an atmVclConnKind value of pvc and an atmVclOperStatus with a value other than up.

BGP4-MIB

The BGP4-MIB (RFC 1657) provides access to information related to the implementation of the Border Gateway Protocol (BGP). The MIB provides:

- BGP configuration information
- Information about BGP peers and messages exchanged with them
- Information about advertised networks

This table lists the tables associated with this MIB.

Table 1-11 BGP4-MIB Tables and Descriptions

Name	Description
bgpPeerTable	BGP peer table. This table contains one entry per BGP peer and information about the connections with BGP peers.
bgpRcvdPathAttrTable	BGP Received Path Attribute Table contains information about paths to destination networks received from all peers running BGP version 3 or fewer. This table is not supported.
bgp4PathAttrTable	BGP-4 Received Path Attribute Table contains information about paths to destination networks received from all BGP4 peers.
bgpEstablished	This object is deprecated in favour of bgpEstablishedNotification from Release 4.1.0.
bgpBackwardTransition	This object is deprecated in favour of bgpBackwardTransNotification from Release 4.1.0.

MIB Constraints

This table lists the constraints that the router places on objects in the BGP4-MIB.

Table 1-12 BGP4-MIB Constraints

MIB Object	Notes
bgpRcvdPathAttrTable	Not supported

BRIDGE-MIB

The BRIDGE-MIB contains objects to manage MAC bridges between LAN segments, as defined by the IEEE 802.1D-1990 standard. This MIB is extracted from RFC 1493 and is intended for use with network management protocols in TCP/IP-based internets.

This table lists the tables associated with this MIB.

Table 1-13 BRIDGE-MIB Tables and Descriptions

Name	Description
dot1dBasePortTable	Table that contains generic information about every port that is associated with this bridge. Transparent, source-route, and srt ports are included.
dot1dStpPortTable	Table that contains port-specific information for the Spanning Tree Protocol
dot1dTpFdbTable	Table that contains information about unicast entries for which the bridge has forwarding and filtering information. This information is used by the transparent bridging function in determining how to propagate a received frame.
dot1dTpPortTable	Table that contains information about every port that is associated with this transparent bridge.
dot1dStaticTable	Table containing filtering information configured into the bridge by (local or network) management specifying the set of ports to which frames received from specific ports and containing specific destination addresses are allowed to be forwarded. The value of zero in this table, as the port number from which frames with a specific destination address are received, is used to specify all ports for which there is no specific entry in this table for that particular destination address. Entries are valid for unicast and for group and broadcast addresses.

MIB Constraints

This table lists the constraints that the router places on objects in the BRIDGE-MIB. For detailed definitions of MIB objects, see the MIB. This MIB only supports managing two types of bridges (CE and VPLS bridges).

Note Set Operation on BRIDGE-MIB objects is not supported.

Table 1-14 BRIDGE-MIB Constraints

MIB Object	Notes
dot1dStp Subtree Objects	Not supported for VPLS Bridges
newRoot	Not supported for VPLS Bridges
TCN Traps	Not supported for VPLS Bridges

CISCO-ATM-EXT-MIB

The CISCO-ATM-EXT-MIB is an extension to the Cisco ATM MIB module for managing ATM implementations.

This table lists the tables associated with this MIB.

Table 1-15 ***CISCO-ATM-EXT-MIB Tables and Descriptions***

Name	Description
cAal5VccExtTable	This table contains AAL5 VCC performance parameters beyond that provided by cAal5VccEntry.
catmxVclOamTable	This table contains VCL[1] Oam configuration and state information. This table augments the atmVclTable.

[1] VCL = Virtual Channel Link

CISCO-ATM-QOS-MIB

The CISCO-ATM-QOS-MIB is created to provide ATM QoS information in the following areas:

- Traffic shaping on a per-VC basis
- Traffic shaping on a per-VP basis
- Per-VC queuing or buffering

Although the initial requirements of the MIB are driven to support the Cisco XR 12000 Series router TAZ line card, CISCO-ATM-QOS-MIB is designed as a generic MIB to support ATM interfaces cross all platforms.

This table lists the tables associated with this MIB.

Table 1-16 ***CISCO-ATM-QOS-MIB Tables and Description***

Name	Description
caqVccParamsTable	This table is defined to provide QoS information for each active ATM VC existing on the interface.
caqVpcParamsTable	This table is defined to provide QoS information for each active ATM VP existing on the interface.
caqQueuingParamsTable	This table provides queuing related information for a VC existing on an ATM interface.
caqQueuingParamsClassTable	This table provides queuing information for all queuing classes associating with a VC.

CISCO-BGP4-MIB

The CISCO-BGP4-MIB provides access to information related to the implementation of the Border Gateway Protocol (BGP). The MIB provides:

- BGP configuration information
- Information about BGP peers and messages exchanged with them
- Information about advertised networks

This table lists the tables associated with this MIB.

Table 1-17 ***CISCO-BGP4-MIB Tables and Descriptions***

Name	Description
cbgpRouteTable	This table contains information about routes to destination networks from all BGP4 peers. Because BGP4 can carry routes for multiple Network Layer protocols, this table has the AFI[1] of the Network Layer protocol as the first index. Further for a given AFI, routes carried by BGP4 are distinguished based on SAFI. Hence, that is used as the second index. Conceptually there is a separate Loc-RIB maintained by the BGP speaker for each combination of AFI and SAFI supported by it.
cbgpPeerTable	BGP peer table. This table contains, one entry per BGP peer, information about the connections with BGP peers.
cbgpPeerCapsTable	This table contains the capabilities that are supported by a peer. Capabilities of a peer are received during BGP connection establishment. Values corresponding to each received capability are stored in this table. When a new capability is received, this table is updated with a new entry. When an existing capability is not received during the latest connection establishment, the corresponding entry is deleted from the table.
cbgpPeerAddrFamilyTable	This table contains information related to address families supported by a peer. Supported address families of a peer are known during BGP connection establishment. When a new supported address family is known, this table is updated with a new entry. When an address family is not supported any more, corresponding entry is deleted from the table.
cbgpPeerAddrFamilyPrefixTable	This table contains prefix related information related to address families supported by a peer. Supported address families of a peer are known during BGP connection establishment. When a new supported address family is known, this table is updated with a new entry. When an address family is not supported any more, corresponding entry is deleted from the table.

[1] AFI = Address Family Identifiers

CISCO-BGP-POLICY-ACCOUNTING-MIB

The CISCO-BGP-POLICY-ACCOUNTING-MIB describes BGP policy based accounting information. Support is provided for both source and destination IP address based statistics for ingress and egress traffic.

Note CISCO-BGP-POLICY-ACCOUNTING-MIB support is in the context of IPv4 traffic. This MIB is not supported for IPv6.

This table lists the tables associated with this MIB.

Table 1-18 ***CISCO-BGP-POLICY-ACCOUNTING-MIB Tables and Descriptions***

Name	Description
cbpAcctTable	cbpAcctTable provides statistics about ingress and egress traffic on an interface. This data could be used for purposes like billing.

CISCO-BULK-FILE-MIB

The CISCO-BULK-FILE-MIB contains objects to create and delete SNMP data bulk files for file transfer.

This table lists the tables associated with this MIB.

Table 1-19 CISCO-BULK-FILE-MIB Tables and Descriptions

Name	Description
cbfDefineFileTable	Table of bulk file definition and creation controls
cbfDefineObjectTable	Table of objects to go in bulk files
cbfStatusFileTable	Table of bulk file status

MIB Constraints

This table lists the constraints that the router places on objects in the CISCO-BULK-FILE-MIB.

Table 1-20 CISCO-BULK-FILE-MIB Constraints

MIB Object	Notes
cbfDefineFileTable	
cbfDefinedFileStorage	Only *permanent* and *volatile* type of file storage is supported, *ephemeral* is not supported.
cbfDefinedFileFormat	Only *bulkBinary* and *bulkASCII* file formats are supported. standardBER, variantBERWithChksum and variantBinWithChksum are not supported.

Note Bulk file operation reuses repeated get operations, it is not an optimized processing path.

Note The cbfDefineFileTable has objects that are required for defining a bulk file and for controlling its creation. The cbfDefineObjectTable has information the contents (SNMP data) that go into the bulk file. When an entry in the
cbfDefineFileTable and its corresponding entries in the cbfDefineObjectTable are active, then cbfDefineFileNow can be set to create. This causes a bulkFile to be created as defined in cbfDefineFileTable and it creates an entry in the cbfStatusFileTable.

CISCO-CDP-MIB

The CISCO-CDP-MIB module manages the Cisco Discovery Protocol in Cisco devices.

This table lists the tables associated with this MIB.

Table 1-21 CISCO-CDP-MIB Tables and Descriptions

Name	Description
cdpInterfaceTable	(conceptual) Table containing the status of CDP on the device interfaces.
cdpInterfaceExtTable	This table contains the additional CDP configuration on the interface of the device. This table is not supported.
cdpCacheTable	(conceptual) Table containing the cached information obtained via receiving CDP messages.
cdpCtAddressTable	(conceptual) Table containing the list of network-layer addresses of a neighbor interface, as reported in the Address TLV of the most recently received CDP message. The first address included in the Address TLV is saved in cdpCacheAddress. This table contains the remainder of the addresses in the Address TLV. This table is not supported.

MIB Constraints

This table lists the constraints that the router places on objects in the CISCO-CDP-MIB. For detailed definitions of MIB objects, see the MIB.

Table 1-22 CISCO-CDP-MIB Constraints

MIB Object	Notes
cdpInterfaceExtTable	Not supported
cdpCtAddressTable	Not supported

CISCO-CLASS-BASED-QOS-MIB

The CISCO-CLASS-BASED-QOS-MIB provides read access to Quality of Service (QoS) configuration information and statistics for Cisco platforms that support the modular Quality of Service command-line interface (QoS CLI).

To understand how to navigate the CISCO-CLASS-BASED-QOS-MIB tables, it is important to understand the relationship among different QoS objects. QoS objects consists of:

- Match statement–Specific match criteria to identify packets for classification purposes.
- Class map–User-defined traffic class that contains one or more match statements used to classify packets into different categories.
- Feature action–Action taken on classified traffic. Features include police, traffic shaping, queueing, random detect, and packet marking. After the traffic is classified actions are applied to packets matching each traffic class.
- Policy map–User-defined policy that associates QoS feature actions to user-defined class maps as policy maps can have multiple class maps.
- Service policy–Policy map that has been attached to an interface.
- The MIB uses the following indices to identify QoS features and distinguish among instances of those features:
- cbQosObjectsIndex–Identifies each QoS feature on the router.
- cbQoSConfigIndex–Identifies a type of QoS configuration. This index is shared by QoS objects that have identical configurations.
- cbQosPolicyIndex–Uniquely identifies a service policy. QoS MIB information is stored in:
- Configuration instances–Includes all class maps, policy maps, match statements, and feature action configuration parameters. Might have multiple identical instances. Multiple instances of the same QoS feature share a single configuration object, which is identified by cbQosConfigIndex.
- Runtime Statistics instances–Includes summary counts and rates by traffic class before and after any configured QoS policies are enforced. In addition, detailed feature-specific statistics are available for select Policy Map features. Each has a unique run-time instance. Run-time instances of QoS objects are each assigned a unique identifier (cbQosObjectsIndex) to distinguish among multiple objects with matching configurations.

This table lists the tables associated with this MIB.

Table 1-23 CISCO-CLASS-BASED-QOS-MIB Tables and Descriptions

Name	Description
cbQosQueueingClassCfgTable	This table specifies the configuration information for weighted queue limit action per IP precedence basis.
cbQosMeasureIPSLACfgTable	Not supported.
cbQosServicePolicyTable	This table describes the logical interfaces or media types and the policymap that are attached to it.
cbQosInterfacePolicyTable	Not supported.
cbQosIPHCCfgTable	This table specifies the IP header compression configuration information.
cbQosFrameRelayPolicyTable	Not supported.
cbQosATMPVCPolicyTable	Not supported.
cbQosObjectsTable	This table specifies QoS objects (classmap, policymap, match statements, and actions) hierarchy. This table also provides relationship between each PolicyIndex/ ObjectsIndex pair and the ConfigIndex. ConfigIndex is essential for querying any configuration tables.
cbQosPolicyMapCfgTable	This table specifies Policymap configuration information.
cbQosCMCfgTable	This table specifies ClassMap configuration information.
cbQosMatchStmtCfgTable	This table specifies the match statement configuration information.
cbQosQueueingCfgTable	This table specifies Queueing Action configuration information.
cbQosREDCfgTable	This table specifies WRED Action configuration information.
cbQosREDClassCfgTable	This table specifies WRED Action configuration information on a per IP precedence basis.
cbQosPoliceCfgTable	This table specifies Police Action configuration information.
cbQosPoliceActionCfgTable	This table specifies Police Action configuration information.
cbQosTSCfgTable	This table specifies traffic-shaping Action configuration information.
cbQosSetCfgTable	This table specifies Packet Marking Action configuration information.
cbQosCMStatsTable	This table specifies ClassMap related Statistical information.
cbQosMatchStmtStatsTable	Not supported.
cbQosNoBufferDropTable	Not supported.
cbQosPoliceStatsTable	This table specifies Police Action related Statistical information.
cbQosQueueingStatsTable	This table specifies Queueing Action related Statistical information.
cbQosTSStatsTable	This table specifies traffic-shaping Action related Statistical information
cbQosREDClassStatsTable	This table specifies per Precedence WRED Action related Statistical information.
cbQosIPHCCfgTable	This table specifies IP Header Compression configuration information.
cbQosIPHCStatsTable	This table specifies IP Header Compression Statistical information.
cbQosSetStatsTable	Not supported.
cbQosPoliceColorStatsTable	This table specifies Police Action related statistical information for two rate color aware marker.
cbQosTableMapCfgTable	Not supported.

Table 1-23 CISCO-CLASS-BASED-QOS-MIB Tables and Descriptions

Name	**Description**
cbQosTableMapValueCfgTable	Not supported.
cbQosTableMapSetCfgTable	Not supported.
cbQosEBCfgTable	Not supported.
cbQosEBStatsTable	Not supported.
cbQosC3plAccountCfgTable	Not supported.
cbQosC3plAccountStatsTable	Not supported.

MIB Constraints

This table lists the constraints on objects in the CISCO-CLASS-BASED-QOS-MIB. For detailed definitions of MIB objects, see the MIB.

Table 1-24 CISCO-CLASS-BASED-QOS-MIB Constraints

MIB Object	**Notes**
cbQosATMPVCPolicyTable	Not supported
cbQosC3plAccountCfgTable	Not supported on XR
cbQosC3plAccountStatsTable	Not supported on XR
cbQosCMStatsTable	**MIB Object**
■ Lack of SRAM buffers, count is negligible.	
■ Lack of SRAM buffers, count is negligible.	
■ Lack of SRAM buffers, count is negligible.	
cbQosEBCfgTable	Not supported in QoS on XR
cbQosEBStatsTable	Not supported
cbQosEVCGroup	Not supported
cbQosFrameRelayPolicyTable	Not supported
cbQosInterfacePolicyTable	Not supported
cbQosIPHCStatsTable	Only RTP supported on XR.
cbQosMeasureIPSLACfgTable	Not supported on XR
cbQosMatchStmtStatsTable	
■ CbQosMatchPrePolicyPktOverflow	Not supported
■ CbQosMatchPrePolicyPkt	Not supported

Table 1-24 CISCO-CLASS-BASED-QOS-MIB Constraints

MIB Object	**Notes**
■ CbQosMatchPrePolicyPkt64	Not supported
■ CbQosMatchPrePolicyByteOverflow	Not supported
■ CbQosMatchPrePolicyByte	Not supported
■ CbQosMatchPrePolicyByte64	Not supported
■ CbQosMatchPrePolicyBitRate	Not supported
MIB Object	**Notes**
cbQosNoBufferDropTable	Not supported
cbQosPoliceCfgTable	
■ cbQosPoliceCfgConformAction	Deprecated and defined in cbQosPoliceActionCfgTable
■ cbQosPoliceCfgConformSetValue	Deprecated and defined in cbQosPoliceActionCfgTable
■ cbQosPoliceCfgExceedAction	Deprecated and defined in cbQosPoliceActionCfgTable
■ cbQosPoliceCfgExceedSetValue	Deprecated and defined in cbQosPoliceActionCfgTable
■ cbQosPoliceCfgViolateAction	Deprecated and defined in cbQosPoliceActionCfgTable
■ cbQosPoliceCfgViolateSetValue	Deprecated and defined in cbQosPoliceActionCfgTable
cbQosQueueingCfgTable	
■ cbQosQueueingCfgFlowEnabled	Not supported
■ cbQosQueueingCfgAggregateQSize	Deprecated by cbQosQueueingCfgAggregateQLimit
■ cbQosQueueingCfgDynamicQNumber	Not supported
■ cbQosQueueingCfgPrioBurstSize	Not supported
cbQosQueueingClassCfgTable	Not supported
cbQosREDCfgTable	
■ cbQosREDCfgExponWeight	Not supported on XR
■ cbQosREDCfgMeanQSize	Replaced by cbQosREDMeanQsize
■ cbQosREDCfgECNEnabled	Not supported
cbQosREDClassCfgTable	

Table 1-24 CISCO-CLASS-BASED-QOS-MIB Constraints

MIB Object	**Notes**
■ cbQosREDCfgMinThreshold	Deprecated by cbQosREDClassCfgMInThreshold. For XR, many objects from cbQosRedCfg are now available via cbQosREDClassCfg
■ cbQosREDCfgMaxThreshold	Deprecated by cbQosREDClassCfgMInThreshold. For XR, many objects from cbQosRedCfg are now available via cbQosREDClassCfg
cbQosREDClassStatsTable	
■ CbQosREDMeanQSizeUnits	Not supported
■ CbQosREDMeanQSize	Not supported
cbQosServicePolicyTable	
cbQosEntityIndex	Not supported
cbQosVlanIndex	Not supported
cbQosSetStatsTable	Marking statistics are not supported on XR
cbQosTableMapCfgTable	Not supported on XR
cbQosTableMapValueCfgTable	Not supported
cbQosTableMapSetCfgTable	Not supported
cbQosTrafficShapingDelayCountersGroup	Not supported
cbQosTSCfgTable	
■ CbQosTSCfgBurstsize	Not supported
MIB Object	**Notes**
■ CbQosTSCfgAdaptiveEnabled	Not supported
■ CbQosTSCfgAdaptiveRate	Not supported
cbQosCMStatsTable	
■ cbQosCMNoBufDropPktOverflow	Not supported
■ cbQosCMNoBufDropPkt	Not supported
■ cbQosCMNoBufDropPkt64	Not supported
■ cbQosCMFragmentPktOverflow	Not supported
■ cbQosCMFragmentPkt	Not supported
■ cbQosCMFragmentPkt64	Not supported
■ cbQosCMFragmentByteOverflow	Not supported
■ cbQosCMFragmentByte	Not supported

Table 1-24 CISCO-CLASS-BASED-QOS-MIB Constraints

MIB Object	Notes
■ cbQosCMFragmentByte64	Not supported
■ cbQosCMPrePolicyBitRate64	Not supported till Release 4.3.x.
■ cbQosCMPostPolicyBitRate64	Not supported till Release 4.3.x.
■ cbQosCMDropBitRate64	Not supported till Release 4.3.x.
cbQosPoliceStatsTable	
■ cbQosPoliceConformedBitRate64	Not supported till Release 4.3.x.
■ cbQosPoliceExceededBitRate64	Not supported till Release 4.3.x.
■ cbQosPoliceViolatedBitRate64	Not supported till Release 4.3.x.
cbQosTSStatsTable	
■ CbQosTSStatsCurrentQSize	Not supported

CISCO-CONFIG-COPY-MIB

The CISCO-CONFIG-COPY-MIB contains objects to copy configuration files on the router. For example, the MIB enables the SNMP agent to copy:

- Configuration files to and from the network
- Running configuration to the startup configuration and startup to running
- Startup or running configuration files to and from a local Cisco IOS XR Software file system

This table lists the tables associated with this MIB.

Table 1-25 CISCO-CONFIG-COPY-MIB Tables and Descriptions

Name	Description
ccCopyTable	Table of config-copy requests.
ccCopyErrorTable	Table containing information about the failure cause of the config copy operation. An entry is created only when the value of ccCopyState changes to 'failed' for a config copy operation. Not all combinations of ccCopySourceFileType and ccCopyDestFileType need to be supported. For example, an implementation may choose to support only the following combination: ccCopySourceFileType = 'runningConfig' ccCopyDestFileType ='fabricStartupConfig'. In this case where a fabric wide config copy operation is being performed, for example by selecting ccCopyDestFileType value to be 'fabricStartupConfig', it is possible that the fabric could have more than one device. In such cases this table would have one entry for each device in the fabric. In this case even if the operation succeeded in one device and failed in another, the operation as such has failed, so the global state represented by ccCopyState 'failed', but for the device on which it was success, ccCopyErrorDescription would have the distinguished value, 'success'. After the config copy operation finishes and if an entry gets instantiated, the management station should retrieve the values of the

Table 1-25 ***CISCO-CONFIG-COPY-MIB Tables and Descriptions***

Name	Description
	status objects of interest. After an entry in ccCopyTable is deleted by management station, all the corresponding entries with the same ccCopyIndex in this table are also deleted. To prevent old entries from clogging the table, entries age out at the same time as the corresponding entry with same ccCopyIndex in ccCopyTable ages out.

CISCO-CONFIG-MAN-MIB

The CISCO-CONFIG-MAN-MIB contains objects to track and save changes to the router configuration. The MIB represents a model of the configuration data that exists in the router and in the peripheral devices. Its main purpose is to report changes to the running configuration through the SNMP notification ciscoConfigManEvent.

This table lists the tables associated with this MIB.

Table 1-26 ***CISCO-CONFIG-MAN-MIB Tables and Descriptions***

Name	Description
ccmHistoryEventTable	Table of configuration events on this router
ccmCLIHistoryCommandTable	Table of CLI commands that took effect during configuration events

CISCO-CONTEXT-MAPPING-MIB

The CISCO-CONTEXT-MAPPING-MIB provides option to associate an SNMP context to a feature package group. This MIB allows manageability of license MIB objects specific to a feature package group.

A single SNMP agent sometimes needs to support multiple instances of the same MIB module, and does so through the use of multiple SNMP contexts. This typically occurs because the technology has evolved to have extra dimensions; that is, one or more extra data value, identifier value or both which are different in the different contexts, but were not defined in INDEX clauses of the original MIB module. In such cases, network management applications need to know the specific data or identifier values in each context, and this MIB module provides mapping tables which contain that information.

Within a network there can be multiple VPNs configured using Virtual Routing and Forwarding Instances (VRFs). Within a VPN there can be multiple topologies when Multi-topology Routing (MTR) is used. Also, Interior Gateway Protocols (IGPs) can have multiple protocol instances running on the device. A network can have multiple broadcast domains configured using Bridge Domain Identifiers.

With MTR routing, VRFs, and Bridge domains, a router now needs to support multiple instances of several existing MIB modules, and this can be achieved if the SNMP agent of the router provides access to each instance of the same MIB module via a different SNMP context (see RFC 3411). For MTR routing, VRFs, and Bridge domains, a different SNMP context is needed depending on one or more of the following: the VRF, the topology-identifier, the routing protocol instance, and the bridge domain identifier. In other words, the management information of the router can be accessed through multiple SNMP contexts where each such context represents a specific VRF, a specific topology-identifier, a specific routing protocol instance or a bridge domain identifier. This MIB module provides a mapping of each such SNMP context to the corresponding VRF, the corresponding topology, the corresponding routing protocol instance, and the corresponding bridge domain identifier. Some SNMP contexts are independent of VRFs, independent of a topology, independent of a routing protocol instance, or independent of a bridge domain and in such a case, the mapping is to the zero length string.

With the Cisco package licensing strategy, the features available in the image are grouped into multiple packages and each package can be managed to operate at different feature levels based on the available license.

This table lists the tables associated with this MIB.

Table 1-27 CISCO-CONTEXT-MAPPING-MIB Tables and Descriptions

Name	Description
cContextMappingTable	This table contains information on which cContextMappingVacmContextName is mapped to which VRF, topology, and routing protocol instance. This table is indexed by SNMP VACM context. Configuring a row in this table for an SNMP context does not require that the context be already defined; that is, a row can be created in this table for a context before the corresponding row is created in RFC 3415 vacmContextTable. To create a row in this table, a manager must set cContextMappingRowStatus to either 'createAndGo' or 'createAndWait'. To delete a row in this table, a manager must set cContextMappingRowStatus to 'destroy'.
cContextMappingBridgeDomainTable	This table contains information on which cContextMappingVacmContextName is mapped to which bridge domain. A Bridge Domain is one of the means by which it is possible to define an Ethernet broadcast domain on a bridging device. A network can have multiple broadcast domains configured. This table helps the network management personnel to find out the details of various broadcast domains configured in the network. An entry need to exist in cContextMappingTable, to create an entry in this table.
cContextMappingBridgeInstanceTable	This table contains information on mapping between cContextMappingVacmContextName and bridge instance. Bridge instance is an instance of a physical or logical bridge that has unique bridge-id. If an entry is deleted from cContextMappingTable, the corresponding entry in this table also gets deleted. If an entry needs to be created in this table, the corresponding entry must exist in cContextMappingTable.
cContextMappingLicenseGroupTable	This table contains information on which cContextMappingVacmContextName is mapped to a License Group. Group level licensing is used where each Technology Package is enabled via a License.

CISCO-DS3-MIB

The CISCO-DS3-MIB describes DS3 line objects. This is an extension to the standard DS3 MIB (RFC 2496).

This table lists the tables associated with this MIB.

Table 1-28 CISCO-DS3-MIB Tables and Descriptions

Name	Description
cds3ConfigTable	This table has objects for configuring a T3/E3 line.
cds3AlarmConfigTable	This table contains the parameters associated with detecting and declaring alarms for the interface. The parameters include severity of alarm, alarm

Table 1-28 ***CISCO-DS3-MIB Tables and Descriptions***

Name	Description
	integration parameters, and 15-minute and 24-hour thresholds.
cds3StatsTable	T3/E3 Statistics table. This table maintains the number of times the line encountered LOS[1], LOF[2], AIS[3], RAI[4], CCV[5], FE[6], from the time it is up. Line fails and goes down. When the line is brought back up again by the user the error statistics are cleared.
cds3AlarmConfigPlcpTable	ATM interface PLCP alarm configuration table. PLCP is a sublayer over the DS3 interface, that carries ATM cells.
cds3AlarmPlcpTable	Plcp interface alarm table. This table maintains the CV,ES, SES, SEFS and UAS for DS3 line with Plcp framing selected. See RFC 2496 for description of these various error statistics.
cds3AlarmPlcpTable	Plcp interface alarm table. This table maintains the CV,ES, SES, SEFS and UAS for DS3 line with Plcp framing selected. See RFC 2496 for description of these various error statistics.
cds3PlcpStatsTable	T3 Plcp Statistics table. This table maintains the errors encountered by the T3 line with Plcp frame format selected, from the time the line is up. Line fails and goes down. When the line is brought back up again by the user after eliminating the error conditions, the statistics are cleared.
cds3PlcpStatsTable	T3 Plcp Statistics table. This table maintains the errors encountered by the T3 line with Plcp frame format selected, from the time the line is up. Line fails and goes down. When the line is brought back up again by the user after eliminating the error conditions, the statistics are cleared.
cds3IntervalTable	DS3 interface interval table.
cds3Current24HrTable	DS3 interface current 24-hour table. This table contains counters for current 24-hour interval. Threshold on this counters are configured through cds3AlarmConfigTable table. 24-hour interval is aligned to wall clock.
cds3Previous24HrTable	DS3 interface previous 24-hour table. This table contains counters for previous 24-hour interval. Implementation of this table is optional.

[1] LOS = loss of signal
[2] LOF = out of frame
[3] AIS = alarm indication signals
[4] RAI = remote alarm indications
[5] CCV = C-bit coding violations
[6] FE = framing errors

MIB Constraints

This table lists the constraints on objects in the CISCO-DS3-MIB. For detailed definitions of MIB objects, see the MIB.

Table 1-29 ***CISCO-DS3-MIB Constraints***

MIB Object	Notes
cds3LineAIScBitsCheck	Not supported
cds3LineRcvFEACValidation	Not supported
cds3LineOOFCriteria	Not supported
cds3TraceAlarm	Not supported

Table 1-29 CISCO-DS3-MIB Constraints

MIB Object	Notes
cds3InternalEqualizer	Not supported

CISCO-ENHANCED-IMAGE-MIB

The CISCO-ENHANCED-IMAGE-MIB provides information about events running on the system. This MIB has Image table containing the following information related to the running the Cisco IOS XR software image:

- Entity index
- Image name
- Family
- Feature set
- Version
- Media
- Description

Note Only ceImageTable is supported in this MIB.

This table lists the tables associated with this MIB.

Table 1-30 CISCO-ENHANCED-IMAGE-MIB Tables and Descriptions

Name	Description
ceImageTable	This table provides information describing the executing image. For modular operating systems this table provides base image or MBI.
ceImageLocationTable	This table is applicable to modular operating systems. A location describes where on the file system the installed software is placed. This table consists of list of all locations along with status of image at that location. ceImageLocationRunningStatus is true only for the location from where system is currently operational. The agent may add entries to this table when a new image is installed on the system. The agent may delete entries from this table when an image has been removed from the system.
ceImageInstallableTable	This table specifies a list of software drivers installed on the system. This table is applicable to operating systems which support installables. A modular operating system can consist of base image or MBI and installables. The value of ceImageLocationIndex can be used as index to retrieve installables installed at a particular location. Every image has a table of installables. Entries are added in this table when an installable is installed on the image. Entries are deleted from this table when installables are removed or rolled back from the image.
ceImageTagTable	A tag is a virtual label placed by a user that indicates a point deemed to be stable. It can be used to rollback to a system after an install that negatively impacts the functionality of the system. It gives point in system where user can go back to, to remove drivers installed after that point of time. When a tag is placed on an image, an entry appears in this table. An entry is removed from this table when tag is removed from the system. The value of ceImageLocationIndex is used as index to get all the tags that are placed on the image at this location.

CISCO-ENHANCED-MEMPOOL-MIB

The CISCO-ENHANCED-MEMPOOL-MIB contains objects to monitor memory pools on all of the physical entities on a managed system. It represents the different types of memory pools that may be present in a managed device. Memory use information is provided to users at three different intervals of time: 1 minute, 5 minutes, and 10 minutes. Memory pools can be categorized into two groups, predefined pools and dynamic pools. The following pool types are currently predefined:

- 1:Processor memory
- 2:I/O memory
- 3:PCI memory
- 4:Fast memory
- 5:Multibus memory

Dynamic pools have a pool type value greater than any of the predefined types listed above. Only the processor pool is required to be supported by all devices. Support for other pool types is dependent on the device being managed.

This table lists the tables associated with this MIB.

Table 1-31 CISCO-ENHANCED-MEMPOOL-MIB Tables and Descriptions

Name	**Description**
cempMemPoolTable	Table of memory pool monitoring entries for all physical entities on a managed system.
cempMemBufferPoolTable	Entries in this table define entities (buffer pools in this case) which are contained in an entity (memory pool) defined by an entry from cempMemPoolTable.
cempMemBufferCachePoolTabl e	Table that lists the cache buffer pools configured on a managed system. ■ To provide a noticeable performance boost, Cache Pool can be used. A Cache Pool is effectively a lookaside list of free buffers that can be accessed quickly. Cache Pool is tied to Buffer Pool. ■ Cache pools can optionally have a threshold value on the number of cache buffers used in a pool. This can provide flow control management by having an implementation specific approach such as invoking a vector when pool cache rises above the optional threshold set for it on creation.

MIB Constraints

This table lists the constraints on objects in the CISCO-ENHANCED-MEMPOOL-MIB. For detailed definitions of MIB objects, see the MIB.

Table 1-32 CISCO-ENHANCED-MEMPOOL-MIB Constraints

Name	**Description**
ciscoMemoryPoolTable	
cempMemPoolType	Values are: ■ processorMemory (2) ■ ioMemory (3)

Table 1-32 CISCO-ENHANCED-MEMPOOL-MIB Constraints

Name	Description
cempMemPoolAlternate	Always 0
cempMemPoolPlatformMemory	Always 0

CISCO-ENTITY-ASSET-MIB

The CISCO-ENTITY-ASSET-MIB provides asset tracking information for the physical components in the ENTITY-MIB (RFC 2737) entPhysicalTable.

The ceAssetTable contains an entry (ceAssetEntry) for each physical component on the router. Each entry provides information about the component, such as its orderable part number, serial number, hardware revision, manufacturing assembly number, and manufacturing revision.

Most physical components are programmed with a standard Cisco generic Identification Programmable Read-Only Memory (IDPROM) value that specifies asset information for the component. If possible, the MIB accesses the IDPROM information of the component.

This table lists the tables associated with this MIB.

Table 1-33 CISCO-ENTITY-ASSET-MIB Tables and Descriptions

Name	Description
ceAssetTable	This table lists the orderable part number, serial number, hardware revision, manufacturing assembly number and revision, firmwareID and revision if any, and softwareID and revision if any, of relevant entities listed in the ENTITY-MIB entPhysicalTable. Entities for which none of this data is available are not listed in this table. This is a sparse table, so some of these variables may not exist for a particular entity at a particular time. For example, a powered-off module does not have softwareID and revision; a power-supply would probably never have firmware or software information. Although the data may have other items encoded in it (for example manufacturing-date in the serial number) treat all data items as monolithic. Do not decompose them or parse them. Use only string equals and unequals operations on them.

This table gives more information on the objects associated with this MIB.

Table 1-34 CISCO-ENTITY-ASSET-MIB Objects and Value Information

Name	Description
ceAssetMfgAssyNumber	Top-level assembly number stored in IDPROM
ceAssetMfgAssyRevision	This object should reflect the revision of the TAN stored in IDPROM.
ceAssetFirmwareID	This object value should be the same as entPhysicalFirmwareRev of ENTITY-MIB.
ceAssetSoftwareID	This object value should be the same as entPhysicalSoftwareRev of ENTITY-MIB.
ceAssetCLEI	This object should reflect the value of the CLEI stored in the IDPROM supported by the physical entity

MIB Constraints

This table lists the constraints on objects in the CISCO-ENTITY-ASSET-MIB.

Note The current implementation of IOS XR supports only ceAssetGroupRev1 group.

Table 1-35 CISCO-ENTITY-ASSET-MIB Constraints

MIB Object	**Notes**
ceAssetTable	
ceAssetOrderablePartNumber	Not implemented
ceAssetSerialNumber	Not implemented
ceAssetHardwareRevision	Not implemented
ceAssetFirmwareRevision	Not implemented
ceAssetSoftwareRevision	Not implemented
ceAssetAlias	Not implemented
ceAssetTag	Not implemented
ceAssetIsFRU	Not implemented

CISCO-ENTITY-FRU-CONTROL-MIB

The CISCO-ENTITY-FRU-CONTROL-MIB is used to monitor and configure operational status of Field Replaceable Units (FRUs) and other manageable physical entities of the system listed in the Entity-MIB (RFC 2737) entPhysicalTable. FRUs include assemblies such as power supplies, fans, processor modules and interface modules, and so forth.

This table lists the tables associated with this MIB.

Table 1-36 CISCO-ENTITY-FRU-CONTROL-MIB Tables and Descriptions

Name	**Description**
cefcFRUPowerSupplyGroupTable	This table lists the redundancy mode and the operational status of the power supply groups in the system.
cefcFRUPowerStatusTable	This table lists the power-related administrative status and operational status of the manageable components in the system.
cefcFRUPowerSupplyValueTable	This table lists the power capacity of a power FRU in the system, if it provides variable power. Power supplies usually provide either system or inline power. They cannot be controlled by software to dictate how they distribute power. We can also have what are known as variable power supplies. They can provide both system and inline power and can be varied within hardware defined ranges for system and inline limited by a total maximum combined output. They could be configured by the user via CLI or SNMP or be controlled by software internally. This table supplements the information in the cefcFRUPowerStatusTable for power supply FRUs. The cefcFRUCurrent attribute in that table provides the overall current the power supply FRU can provide while this table gives us the individual contribution toward system and inline power.
cefcModuleTable	cefcModuleTable entry lists the operational and administrative status information for ENTITY-MIB entPhysicalTable entries for manageable

Table 1-36 CISCO-ENTITY-FRU-CONTROL-MIB Tables and Descriptions

Name	Description
	components of type PhysicalClass module(9).
cefcIntelliModuleTable	This table sparsely augments the cefcModuleTable (that is, every row in this table corresponds to a row in the cefcModuleTable but not necessarily vice-versa). A cefcIntelliModuleTable entry lists the information specific to intelligent modules which cannot be provided by the cefcModuleTable.
cefcFanTrayStatusTable	This table contains the operational status information for all ENTITY-MIB entPhysicalTable entries which have an entPhysicalClass of 'fan'; specifically, all entPhysicalTable entries which represent either: one physical fan, or a single physical 'fan tray' which is a manufactured (inseparable in the field) combination of multiple fans.
cefcPhysicalTable	This table contains one row per physical entity.
cefcPowerSupplyInputTable	This table contains the power input information for all the power supplies that have entPhysicalTable entries with 'powerSupply' in the entPhysicalClass. The entries are created by the agent at the system power-up or power supply insertion. Entries are deleted by the agent upon power supply removal. The number of entries is determined by the number of power supplies and number of power inputs on the power supply.
cefcPowerSupplyOutputTable	This table contains a list of possible output mode for the power supplies, whose ENTITY-MIB entPhysicalTable entries have an entPhysicalClass of 'powerSupply'. It also indicates which mode is the operational mode within the system.
cefcChassisCoolingTable	This table contains the cooling capacity information of the chassis whose ENTITY-MIB entPhysicalTable entries have an entPhysicalClass of 'chassis'.
cefcFanCoolingTable	This table contains the cooling capacity information of the chassis whose ENTITY-MIB entPhysicalTable entries have an entPhysicalClass of 'chassis'.
cefcModuleCoolingTable	This table contains the cooling requirement for all the manageable components of type entPhysicalClass 'module'.
cefcFanCoolingCapTable	This table contains a list of the possible cooling capacity modes and properties of the fans, whose ENTITY-MIB entPhysicalTable entries have an entPhysicalClass of 'fan'.
cefcConnectorRatingTable	This table contains the connector power ratings of FRUs. Only components with power connector rating management are listed in this table'.
cefcModulePowerConsumptionTable	This table contains the total power consumption information for modules whose ENTITY-MIB entPhysicalTable entries have an entPhysicalClass of 'module'.
cefcModulePowerConsumptionTable	This table contains the total power consumption information for modules whose ENTITY-MIB entPhysicalTable entries have an entPhysicalClass of 'module'.

MIB Constraints

This table lists the constraints that the router places on objects in the CISCO-ENTITY-FRU-CONTROL-MIB.

Table 1-37 CISCO-ENTITY-FRU-CONTROL-MIB Constraints

MIB Object	Notes
cefcModuleTable	

Table 1-37 *CISCO-ENTITY-FRU-CONTROL-MIB Constraints*

MIB Object	Notes
cefcModuleAdminStatus	Set operation not supported
cefcModuleOperStatus	unknown (1) ok (2) failed (7)
cefcModuleResetReason	unknown (1) powerUp (2) manualReset (5)
cefcModuleLastClearConfigTime	Not implemented
cefcModuleResetReasonDescription	Not implemented
cefcModuleStateChangeReasonDescr	Not implemented

CISCO-ENTITY-REDUNDANCY-MIB

The CISCO-ENTITY-REDUNDANCY-MIB management information module supports configuration, control and monitoring of redundancy protection for various kinds of components on Cisco managed devices. It is meant to be generic enough to handle basic redundancy control and monitoring for many types of redundant member components and redundancy architectures as long as there is an Entity MIB entPhysicalIndex and entPhysicalVendorType assigned to each member component. It is designed so that the tables can be augmented in other extension MIBS which build upon this MIB by adding additional objects that may be specific to a particular type of redundancy or member component. This MIB can also be used in cases where some types of redundancy groups and members don’t require explicit user configuration. One example may be redundant fan assemblies. In those cases, the managed system should internally assign group and member indexes, so that it can provide read-only access to the group and member tables. This allows MIB monitoring for these types of redundant entities. The CISCO-ENTITY-REDUNDANCY-MIB is supported from Release 4.2.1 onwards.

MIB Tables

This table lists the tables in CISCO-ENTITY-REDUNDANCY-MIB:

Table 1-38 CISCO-ENTITY-REDUNDANCY-MIB Tables

MIB Table	Description
ceRedunGroupTypesTable	This table lists the basic types of redundancy groups supported on the managed device along with additional information about each group type.
ceRedunVendorTypesTable	This table lists all entPhysicalVendorTypes allowed as members for a specific ceRedunGroupTypeIndex on the managed device, inclusive for all configurable values for ceRedunType, ceRedunScope, ceRedunArch, etc. If the ceRedunGroupDefinitionChanged object changes for a particular ceRedunGroupTypeIndex, then this table may have changed and should be read again. **Note**: Although a specific ceRedunGroupTypeIndex may allow groups of different entPhysicalVendorTypes, managed devices typically enforce all members within a specific group to have the same entPhysicalVendorType.
ceRedunInternalStatesTable	This table allows the managed system to report a read-only list of internal state numbers and the corresponding descriptions which apply for the members of a particular redundancy group type. If the ceRedunGroupDefinitionChanged object changes for a particular ceRedunGroupTypeIndex, then this table may have changed and should be read again.
ceRedunSwitchoverReasonTable	This table allows the managed system to report a read-only list of switchover reason indexes and the corresponding descriptions. If the ceRedunGroupDefinitionChanged object changes for a particular ceRedunGroupTypeIndex, then this table may have changed and should be read again.
ceRedunGroupTable	This table lists group configuration and status objects for a specific redundancy group. However, the members are configured separately in the ceRedunMbrTable.
ceRedunMbrConfigTable	This table lists the group members and generic redundancy objects which are associated with configuring redundancy group members. The switchover granularity should be for one member at a time. In other words if a member is allowed to be an individual port, then switchovers on multi-port linecards would be expected to take place independently for each port on the linecard. But if the members are full linecards, then all ports on the linecard would be expected to switch at the same time.
ceRedunMbrStatusTable	This table lists the redundancy status and other read-only redundancy objects which are associated with redundancy group members. Status associated with member alarm conditions should be reported separately using the CISCO-ENTITY-ALARM-MIB.
ceRedunCommandTable	This table allows switchover commands to be sent to members of configured redundancy groups.

MIB Constraints

This table lists the constraints that the router places on the objects in the CISCO-ENTITY-REDUNDANCY-MIB. For detailed definitions of MIB objects, see the MIB.

Table 1-39 CISCO-ENTITY-REDUNDANCY-MIB Constraints

MIB Object	Notes
ceRedunGroupTypesTable	
ceRedunGroupTypeName	Supported
ceRedunGroupCounts	Supported
ceRedunNextUnusedGroupIndex	Supported
ceRedunMaxMbrsInGroup	Supported
ceRedunUsesGroupName	Supported
ceRedunGroupDefinitionChanged	Supported
ceRedunVendorTypesTable	
ceRedunVendorType	Supported
ceRedunInternalStatesTable	
ceRedunStateCategory	Supported
ceRedunInternalStateDescr	Supported
ceRedunSwitchoverReasonTable	
ceRedunReasonCategory	Supported
ceRedunSwitchoverReasonDescr	Supported
ceRedunGroupTable	
ceRedunGroupString	Supported
ceRedunGroupRedunType	Supported
ceRedunGroupScope	Supported
ceRedunGroupArch	Supported
ceRedunGroupRevert	Supported
ceRedunGroupStorageType	Supported
ceRedunGroupRowStatus	Supported
ceRedunMbrConfigTable	
ceRedunMbrPhysIndex	Supported
ceRedunMbrMode	Supported
ceRedunMbrStorageType	Supported
ceRedunMbrRowStatus	Supported
ceRedunMbrStatusTable	
ceRedunMbrStatusCurrent	Supported
ceRedunMbrProtectingMbr	Supported
ceRedunMbrInternalState	Supported

Table 1-39 ***CISCO-ENTITY-REDUNDANCY-MIB Constraints***

MIB Object	Notes
ceRedunMbrSwitchoverCounts	Supported
ceRedunMbrLastSwitchover	Supported
ceRedunMbrSwitchoverReason	Supported
Scalar Objects	
ceRedunGroupLastChanged	Supported
ceRedunMbrLastChanged	Supported
ceRedunMbrStatusLastChanged	Supported

Notes:

- MIB tables and objects which are not included in the above list are not supported.
- The **ceRedunGroupTable** and **ceRedunMbrConfigTable** have RowStatus objects and they are implemented as read-only. The access for other objects in these tables will be implemented as read-only.

CISCO-ENTITY-SENSOR-MIB

The CISCO-ENTITY-SENSOR-MIB contains objects to monitor the values and thresholds of sensors in ENTITY-MIB entPhysicalTable.

This table lists the tables associated with this MIB.

Table 1-40 ***CISCO-ENTITY-SENSOR-MIB Tables and Descriptions.***

Name	Description
entSensorValueTable	This table lists the type, scale, and present value of a sensor listed in the Entity-MIB entPhysicalTable.
entSensorThresholdTable	This table lists the threshold severity, relation, and comparison value, for a sensor listed in the Entity-MIB entPhysicalTable.

MIB Constraints

This table lists the constraints that the router places on the objects in the CISCO-ENTITY-SENSOR-MIB. For detailed definitions of MIB objects, see the MIB.

Table 1-41 CISCO-ENTITY-SENSOR-MIB Constraints

MIB Object	Notes
entSensorThresholdTable	
entSensorThresholdRelation	Read-only
entSensorThresholdSeverity	Read-only
entSensorThresholdValue	Read-only

MIB Usage Values for Cisco Transceivers

The tables in this section list each type of sensor value represented in the entSensorValueTable and the entSensorThresholdTable.

This table lists CISCO-ENTITY-SENSOR-MIB sensor objects and their usage values for Cisco tranceivers in the entSensor ValueTable.

Table 1-42 ***CISCO***-ENTITY-SENSOR-MIB Usage Values in the entSensorValueTable for Cisco Transceivers

Name	**Description**
Module Temperature Sensor	
entSensorType	celsius (8)
entSensorScale	units (9)
entSensorPrecision	1
entSensorStatus	ok (1)
entSensorValue	Reports most recent measurement seen by the sensor
entSensorValueTimestamp	Value indicates the age of the value reported by entSensorValue object
entSensorValueUpdateRate	Value indicates the rate that the agent updates entSensorValue in sceonds, for example, 60 seconds
Module Voltage Sensor	
entSensorType	volts (DC)4
entSensorScale	units (9)
entSensorPrecision	1
entSensorStatus	ok (1)
entSensorValue	Reports most recent measurement seen by the sensor
entSensorValueTimestamp	Value indicates the age of the value reported by entSensorValue object
entSensorValueUpdateRate	Value indicates the rate that the agent updates entSensorValue in sceonds, for example, 60 seconds
Tx Laser Current Sensor	
entSensorType	amperes (5)
entSensorScale	milli (8)
entSensorPrecision	1
entSensorStatus	ok (1)
entSensorValueTimestamp	Value indicates the age of the value reported by entSensorValue object
entSensorValueUpdateRate	Value indicates the rate that the agent updates entSensorValue in sceonds, for example, 60 seconds
Transmit Power Sensor (Optical Tx) and Receive Power Sensor (Optical Rx)	

Table 1-42 ***CISCO***-ENTITY-SENSOR-MIB Usage Values in the entSensorValueTable for Cisco Transceivers

Name	**Description**
entSensorType	watts (6)
entSensorScale	units (9)
entSensorPrecision	1
entSensorStatus	ok (1)
entSensorValue	Reports most recent measurement seen by the sensor
entSensorValueTimestamp	Value indicates the age of the value reported by entSensorValue object
entSensorValueUpdateRate	Value indicates the rate that the agent updates entSensorValue in sceonds, for example, 60 seconds

Each Cisco transceiver sensor has four threshold values corresponding to the four alarm states listed in Table 1-43. The entSensorValueTable is indexed by both entPhysicalIndex and entSensorThresholdIndex. The Cisco Carrier Routing System entSensorThresholdIndices range from 1 to 4. For N/A, a value of zero is returned.

This table lists the default values for the Cisco transceivers in the entSensorThresholdTable.

Table 1-43 *Default Values* in the entSensorThreshold Table for Cisco Transceivers

MIB Sensor Object	**High Alarm**	**High Warning**	**Low Warning**	**Low Alarm**
Temperature	70.0	60.0	5.00.0	0.0
Voltage	–	–	–	–
Tx Bias Current	80.0	75.0	15.0	10.0
Tx Optical Power	2.0	0.9	-4.0	-9.7
Rx Optical Power	2.0	0.4	-11.9	-15.0

CISCO-ENTITY-STATE-EXT-MIB

The CISCO-ENTITY-STATE-EXT-MIB is a Cisco Specific extension of ENTITY-STATE-MIB specified in RFC 4268. This MIB module is to add objects which provide additional information related to entity states. This MIB define notifications which are generated when a entity undergoes a redundancy switchover.

MIB Tables

This table lists the tables in CISCO-ENTITY-STATE-EXT-MIB:

Table 1-44 ***CISCO-ENTITY-STATE-EXT-MIB Tables***

MIB Table	**Description**
ceStateExtTable	An extension of the entStateTable, defined in ENTITY-STATE-MIB (rfc 4268) providing additional information and control objects for the entities listed in the table.

MIB Constraints

This table lists the constraints that the router places on the objects in the CISCO-ENTITY-STATE-EXT-MIB. For detailed definitions of MIB objects, see the MIB.

Table 1-45 CISCO-ENTITY-STATE-EXT-MIB Constraints

MIB Object	Notes
ceStateExtPrevStandbyState	Not Supported
ceStateExtSwitchoverNotifEnable	Supported
ceStateExtStandbyStatusNotifEnable	Read-only
ceStateExtOperNotifEnable	Read-only
ceStateExtGlobalSwitchoverNotifEnable	Supported
ceStateExtGlobalStandbyStatusNotifEnable	Read-only
ceStateExtGlobalOperNotifEnable	Read-only
ceStateExtStandbySwitchover	Supported
ceStateExtStandbyStatusChange	Not Supported

CISCO-FABRIC-HFR-MIB

This MIB module is used for managing/tracking the Ehanced Benes Fabric entities and/or fabric related configuration, state and statistics information.

Note In a multi chassis system, the output of objects cfhBundlePlane and cfhPlaneId as 0 is a valid output. In CRS, plane identifiers range from 0 to 7. Whenever cfhBundlePlane is polled, bundles belonging to plane-0 will return the output as 0 and polling of cfhPlaneId can also return the output as 0 for plane-0, which should be considered as valid output.

MIB Tables

This table lists the tables in CISCO-FABRIC-HFR-MIB:

Table 1-46 CISCO-FABRIC-HFR-MIB Tables

MIB Table	Description
cfhPlaneTable	This table contains information about fabric plane state and statistics in the managed system.
cfhPlaneStatsTable	This table contains statistics information for the fabric planes in the managed system. Discontinuities in the value of all statistics counters in this table might occur if chassis removal or re-initialization occurs in a managed system. The value of cfhPlaneStatsCounterDiscTime is updated when the counter of fabric plane discontinuity occurs.
cfhBundleTable	This table contains information about fabric bundle state and statistics in the managed system.
cfhBundlePortTable	This table contains one row per fabric bundle port that contains the port state and the aggregation information of traffic and error statistics. The total number of fabric bundle ports is given by the value of cdfhBundlePortNumber.

Table 1-46 CISCO-FABRIC-HFR-MIB Tables

MIB Table	Description
cfhBundlePortStatsTable	This table contains the traffic and error statistics for fabric bundle ports.
cfhCardTable	This table contains a list of fabric connection capable cards for tracking fabric related status, usage and statistics in the system.
cfhCardPlaneTable	This table contains the connectivity information of a card to a fabric plane in a system.

CISCO-FLASH-MIB

The CISCO-FLASH-MIB contains objects to manage flash cards and flash card operations.

This table lists the tables associated with this MIB.

Table 1-47 CISCO-FLASH-MIB Tables and Descriptions

Name	Description
ciscoFlashDeviceTable	Table of Flash device properties for each initialized Flash device. Each Flash device installed in a system is detected, sized, and initialized when the system image boots up. For removable Flash devices, the device properties are dynamically deleted and recreated as the device is removed and inserted. Note that in this case, the newly inserted device may not be the same as the earlier removed one. The ciscoFlashDeviceInitTime object is available for a management station to determine the time at which a device was initialized, and thereby detect the change of a removable device. A removable device that has not been installed also has an entry in this table. This is to let a management station know about a removable device that has been removed. Since a removed device obviously cannot be sized and initialized, the table entry for such a device has ciscoFlashDeviceSize equal to zero, and the following objects have an indeterminate value: ciscoFlashDeviceMinPartitionSize, ciscoFlashDeviceMaxPartitions, ciscoFlashDevicePartitions, and ciscoFlashDeviceChipCount. ciscoFlashDeviceRemovable is true to indicate it is removable.
ciscoFlashChipTable	Table of Flash device chip properties for each initialized Flash device. This table is meant primarily for aiding error diagnosis.
ciscoFlashPartitionTable	Table of flash device partition properties for each initialized flash partition. Whenever there is no explicit partitioning done, a single partition spanning the entire device is assumed to exist. Therefore, there is always at least one partition on a device.
ciscoFlashFileTable	Entry in the table of Flash file properties for each initialized Flash partition. Each entry represents a file and gives details about the file. An entry is indexed using the device number, partition number within the device, and file number within the partition.
ciscoFlashFileByTypeTable	Table of information for files on the manageable flash devices sorted by File Types.

Table 1-47 ***CISCO-FLASH-MIB Tables and Descriptions***

Name	**Description**
ciscoFlashCopyTable	Table of Flash copy operation entries. Each entry represents a Flash copy operation (to or from Flash) that has been initiated.
ciscoFlashPartitioningTable	Table of Flash partitioning operation entries. Each entry represents a Flash partitioning operation that has been initiated.
ciscoFlashMiscOpTable	Table of misc Flash operation entries. Each entry represents a Flash operation that has been initiated.

MIB Constraints

This table lists the constraints on the objects in CISCO-FLASH-MIB.

Table 1-48 ***CISCO***-FLASH-MIB Constraints

MIB Object	**Notes**
ciscoFlashCfgDevInsNotifEnable	Not supported
ciscoFlashCfgDevRemNotifEnable	Not supported
miscOpTable	Verify and erase operations not supported
ciscoFlashPartitioningTable	Not supported
ciscoFlashDeviceInitTime	Not supported
ciscoFlashPhyEntIndex	Not supported
ciscoFlashDeviceSize	Supported, read-only
ciscoFlashDeviceMinPartitionSize	Supported, read-only
ciscoFlashPartitionSize	Supported, read-only
ciscoFlashPartitionFreeSpace	Supported, read-only
ciscoFlashCfgDevInsNotifEnable	Supported, read-only
ciscoFlashCfgDevRemNotifEnable	Supported, read-only
ciscoFlashDeviceCard	Object is deprecated
ciscoFlashDeviceName	Object is deprecated
ciscoFlashDeviceRemovable	Supported, read-only
ciscoFlashDeviceNameExtended	Supported, read-only

CISCO-FRAME-RELAY-MIB

The CISCO-FRAME-RELAY-MIB provides Frame Relay specific information that are either excluded by RFC 1315 (FR DTE MIB) or specific to Cisco products.

This table lists the tables associated with this MIB.

Table 1-49 ***CISCO***-FRAME-RELAY Tables and Descriptions

Name	**Description**
cfrLmiTable	Table of Frame Relay LMI information that are either supplemental to the frDlcmiTable of RFC 1315 or specific to Cisco's implementation.
cfrCircuitTable	Table of descriptive and statistics information that are generic to Frame Relay virtual circuits.
cfrExtCircuitTable	Table of Cisco implementation specific FR circuit information. This is a Cisco extension for the frCircuitTable of RFC 1315.
cfrMapTable	Table of protocols and addresses mapping information of FR virtual circuit.

Table 1-49 ***CISCO***-FRAME-RELAY Tables and Descriptions

Name	**Description**
cfrSvcTable	Table of FR SVC specific, descriptive and statistics information.
cfrElmiTable	Table of Cisco Frame Relay ELMI information that is specific to Cisco implementation.
cfrElmiNeighborTable	Table of Cisco Frame Relay Neighbor ELMI information that is specific to Cisco implementation.
cfrFragTable	Table of Frame Relay Fragmentation information. These are specific to Cisco implementation
cfrConnectionTable	Table of Frame Relay/Frame Relay and Frame Relay/ATM Network/Service Interworking connection information. These are specific to Cisco implementation.

CISCO-FTP-CLIENT-MIB

The CISCO-FTP-CLIENT-MIB contains objects to invoke File Transfer Protocol (FTP) operations for network management. This MIB has no known constraints and all objects are implemented as defined in the MIB.

This table lists the tables associated with this MIB.

Table 1-50 ***CISCO***-FTP-CLIENT-MIB Tables and Descriptions

Name	**Description**
cfcRequestTable	Table of FTP client requests

CISCO-HSRP-EXT-MIB

The CISCO-HSRP-EXT-MIB provides an extension to the CISCO-HSRP-MIB, which defines the Cisco proprietary Hot Standby Routing Protocol (HSRP), defined in RFC 2281. The extensions cover assigning of secondary HSRP ip addresses and modifying priority of an HSRP Group by tracking the operational status of interfaces.

This table lists the tables associated with this MIB.

Table 1-51 ***CISCO-HSRP-EXT-MIB Tables and Descriptions***

Name	**Description**
cHsrpExtIfTrackedTable	Table containing information about tracked interfaces per HSRP group
cHsrpExtSecAddrTable	Table containing information about secondary HSRP IP Addresses per interface and group
cHsrpExtIfTable	HSRP-specific configurations for each physical interface

CISCO-HSRP-MIB

The CISCO-HSRP-MIB provides a means to monitor and configure the Cisco IOS Proprietary Hot Standby Router Protocol (HSRP). Cisco HSRP protocol is defined in RFC 2281.

This table lists the tables associated with this MIB.

Table 1-52 ***CISCO-HSRP-MIB Tables and Descriptions***

Name	**Description**
cHsrpGrpTable	Table containing information on each HSRP group for each interface

CISCO-IETF-BFD-MIB

The CISCO-IETF-BFD-MIB is based on the Internet Draft draft-ietf-bfd-mib-03.txt and draft-ietf-bfd-mib-04.txt. In terms of object syntax and semantics, the content of this Cisco MIB is the same as the corresponding Internet Draft revision. This Cisco MIB was created due to the *subject to change* nature of Internet Drafts. This Cisco MIB may later be deprecated, and the stable RFC, which may replace the Internet Draft, may be implemented in its place.

This table lists the tables associated with this MIB:

Table 1-53 ***CISCO-IETF-BFD-MIB Tables and Descriptions***

Name	**Description**
ciscoBfdSessTable	BFD Session Table describes the BFD sessions
ciscoBfdSessPerfTable	This table specifies BFD Session performance counters
ciscoBfdSessMapTable	BFD Session Mapping Table maps the complex indexing of the BFD sessions to the flat CiscoBfdSessIndexTC used in the ciscoBfdSessTable
ciscoBfdSessDiscMapTable	BFD Session Discriminator Mapping Table maps a local discriminator value to associated BFD sessions' CiscoBfdSessIndexTC used in the ciscoBfdSessTable
ciscoBfdSessIpMapTable	BFD Session IP Mapping Table maps given ciscoBfdSessInterface, ciscoBfdSessAddrType, and ciscoBbfdSessAddr to an associated BFD sessions' CiscoBfdSessIndexTC used in the ciscoBfdSessTable. This table should contain BFD sessions that belong to the following IP type: singleHop(1) and multiHop(2)

MIB Constraints

This table lists the constraints on objects in the CISCO-IETF-BFD-MIB.

Table 1-54 ***CISCO-IETF***-BFD-MIB Constraints

MIB Object	**Notes**
ciscoBfdSessMapTable	Not supported

CISCO-IETF-FRR-MIB

The CISCO-IETF-FRR-MIB contains managed object definitions for MPLS Fast Reroute (FRR) as defined in:Pan, P., Gan, D., Swallow, G., Vasseur, J.Ph., Cooper, D., Atlas, A., Jork, M., Fast Reroute Techniques in RSVP-TE, draft-ietf-mpls-rsvp-lsp-fastreroute- 00.txt, January 2002.

This table lists the tables associated with this MIB.

Table 1-55 ***CISCO-IETF-FRR-MIB Tables and Descriptions***

Name	**Description**
cmplsFrrConstTable	This table shows detour setup constraints
cmplsFrrLogTable	Fast reroute log table records fast reroute events such as protected links going up or down or the FRR feature starting.
mplsFrrOne2OnePlrTable	This table shows the lists of PLRs that initiated detour LSPs, which affect this node.
mplsFrrDetourTable	This table shows all detour LSPs together with their characteristics.
cmplsFrrFacRouteDBTable	mplsFrrFacRouteDBTable provides information about the fast reroute database. Each entry belongs to an interface, protecting backup tunnel and protected tunnel. MPLS interfaces defined on this node are protected by backup tunnels and are indexed by mplsFrrFacRouteProtectedIndex. Backup tunnels defined to protect the tunnels traversing an interface, and are indexed by mplsFrrFacRouteProtectingTunIndex. Note that the tunnel instance index is not required, because it is implied to be 0, which indicates the tunnel head interface for the protecting tunnel. The protecting tunnel is defined to exist on the PLR in the FRR specification. Protected tunnels are the LSPs that traverse the protected link. These LSPs are uniquely identified by: ▪ mplsFrrFacRouteProtectedTunIndex ▪ mplsFrrFacRouteProtectedTunInstance, ▪ mplsFrrFacRouteProtectedTunIngressLSRId ▪ mplsFrrFacRouteProtectedTunEgressLSRId

MIB Constraints

This table lists the constraints on objects in the CISCO-IETF-FRR-MIB.

Table 1-56 ***CISCO***-IETF-FRR-MIB Constraints

MIB Object	**Notes**
mplsFrrOne2OnePlrGroup	Not supported
mplsFrrOne2OnePLRDetour Group	Not supported

CISCO-IETF-IPMROUTE-MIB

The CISCO-IETF-IPMROUTE-MIB is an address family-independent MIB module to manage IP Multicast routing. It is independent of the specific multicast routing protocol. This MIB module is based on RFC 2932 with additional MIB objects to provide address family-independent functionality.

This MIB module contains two scalars and five tables.The tables are:

- IP Multicast Route Table: Containing multicast routing information for IP datagrams sent by a source to the IP multicast groups known to a router.
- IP Multicast Routing Next Hop Table: Containing information on the next hops for the routing IP multicast datagrams.

- IP Multicast Routing Interface Table: Contains multicast routing information specific to interfaces.
- IP Multicast Scope Boundary Table: Containing the boundaries configured for multicast scopes.
- IP Multicast Scope Name Table: Containing names of multicast scope.

Note VRF support was added for this MIB in Cisco IOS Release 4.0.0.

This table lists the tables associated with this MIB.

Table 1-57 ***CISCO-IETF***-IPMROUTE-MIB Tables and Descriptions

Name	**Description**
cIpMRouteTable	(conceptual) Table containing multicast routing information for IP datagrams sent by particular sources to the IP multicast groups known to this router.
cIpMRouteNextHopTable	(conceptual) Table containing information on the next-hops on outgoing interfaces for routing IP multicast datagrams. Each entry is one of a list of next-hops on outgoing interfaces for particular sources sending to a particular multicast group address.
cIpMRouteInterfaceTable	(conceptual) Table containing multicast routing information specific to interfaces.
cIpMRouteBoundaryTable	(conceptual) Table listing the scoped multicast address boundaries of the router.
cIpMRouteScopeNameTable	(conceptual) Table listing the multicast scope names. This table is not supported.

MIB Constraints

This table lists the constraints on objects in the CISCO-IETF-IPMROUTE-MIB.

Table 1-58 ***CISCO-IETF-IPMROUTE-MIB Constraints***

MIB Object	**Notes**
cIpMRouteScopeNameTable	This table is not supported.

CISCO-IETF-MSDP-MIB

The CISCO-IETF-MSDP-MIB is an experimental MIB module for MSDP Management and Monitoring. Version draft-ietf-mboned-msdp-mib-01.txt is ciscoized.

This table lists the tables associated with this MIB.

Table 1-59 ***CISCO-IETF-MSDP-MIB Tables and Descriptions***

Name	**Description**
cMsdpRequestsTable	(conceptual) Table listing group ranges and MSDP peers used when deciding where to send an SA Request message when required. If SA Requests are not enabled, this table may be empty. To choose a peer to whom to send an SA Request for a given group G, the subset of entries in this table whose (cMsdpRequestsPeerType, cMsdpRequestsPeer) tuple represents a peer whose cMsdpPeerState is established are examined. The set is further reduced by examining only those entries for which

Table 1-59 ***CISCO-IETF-MSDP-MIB Tables and Descriptions***

Name	Description
	cMsdpPeerRequestsGroupAddressType equals the address type of G, and the entries with the highest value of cMsdpRequestsGroupPrefix are considered, where the group G falls within the range described by the combination of cMsdpRequestsGroup and cMsdpRequestsGroupPrefix. (This sequence is commonly known as a 'longest-match' lookup.) Finally, if multiple entries remain, the entry with the lowest value of cMsdpRequestsPriority is chosen. The SA Request message is sent to the peer described by this row.
cMsdpPeerTable	(conceptual) Table listing the MSDP speaker's peers.
cMsdpSACacheTable	(conceptual) Table listing the MSDP SA advertisements currently in the MSDP speaker's cache.
cMsdpMeshGroupTable	(conceptual) Table listing MSDP Mesh Group configuration.

CISCO-IETF-PIM-MIB

The CISCO-IETF-PIM-MIB is based on RFC 2934 with additional MIB objects added to make it address family independent MIB. This Cisco MIB was created because of non availability of RFC or an Internet Draft, which can provide address family independent MIB for management of PIM routers. This MIB may later be deprecated with a stable RFC or an Internet Draft.

This table lists the tables associated with this MIB.

Table 1-60 ***CISCO-IETF-PIM-MIB Tables and Descriptions***

Name	Description
cPimIfTable	(conceptual) Table listing the router's PIM interfaces. Along with PIM IGMP or MLD is enabled on all interfaces listed in this table
cPimNbrTable	(conceptual) Table listing the router's PIM neighbors
cPimInetMRouteTable	(conceptual) Table listing PIM-specific information on a subset of the rows of the cIpMRouteTable defined in the IP Multicast MIB
cPimInetMRouteNextHopTable	(conceptual) Table listing PIM-specific information on a subset of the rows of the cIpMRouteNextHopTable defined in the IP Multicast MIB. This table is not supported.
cPimRPMapTable	(conceptual) Table listing PIM information for candidate RPs for IP multicast groups. When the local router is the BSR, this information is obtained from received Candidate-RP-Advertisements. When the local router is not the BSR, this information is obtained from received RP-Set messages. This table is not supported.
cPimCRPTable	(conceptual) Table listing the IP multicast groups for which the local router is to advertise itself as a Candidate-RP when the value of cPimComponentCRPHoldTime is non-zero. If this table is empty, then the local router advertises itself as a Candidate-RP for all groups (providing the value of cPimComponentCRPHoldTime is non- zero). This table is not supported.
cPimComponentTable	(conceptual) Table containing objects specific to a PIM domain. One row exists for each domain to which the router is connected. A PIM-SM domain is defined as an area of the network over which Bootstrap messages are forwarded. Typically, a PIM-SM router is a member of

Table 1-60 CISCO-IETF-PIM-MIB Tables and Descriptions

Name	Description
	exactly one domain. This table also supports routers that may form a border between two PIM-SM domains and do not forward Bootstrap messages between them. This table is not supported.

MIB Constraints

This table lists the constraints on objects in the CISCO-IETF-PIM-MIB.

Table 1-61 CISCO-IETF-PIM-MIB Constraints

MIB Object	Notes
cPimInetMRouteNextHopTable	This table is not supported.
cPimRPMapTable	This table is not supported.
cPimCRPTable	This table is not supported.
cPimComponentTable	This table is not supported.

CISCO-IETF-PIM-EXT-MIB

The CISCO-IETF-PIM-EXT-MIB extends PIM management capabilities defined in CISCO-IETF-PIM-MIB.

This table lists the tables associated with this MIB.

Table 1-62 ***CISCO***-IETF-PIM-EXT-MIB Tables and Descriptions

Name	Description
cpimExtIfTable	(conceptual) Table listing the router's PIM interfaces. IGMP and PIM are enabled on all interfaces listed in this table. This table is augmented to cPimIfTable. This table is not supported.
cpimExtNbrTable	(conceptual) Table listing the router's PIM neighbors. This table is augmented to cPimNbrTable. This table is not supported.
cpimExtNbrSecAddressTable	(conceptual) Table listing the Secondary InetAddresses advertised by each PIM neighbor (on a subset of the rows of the cPimNbrTable defined in CISCO-IETF-PIM-MIB)
cpimExtMRouteTable	(conceptual) Table listing PIM-specific information on a subset of the rows of the cIpMRouteTable defined in the IP Multicast MIB. This table is augmented to cPimInetMRouteTable. This table is not supported.
cpimExtMRouteNextHopTabl e	(conceptual) Table listing PIM-specific information on a subset of the rows of the cIpMRouteNextHopTable defined in the IP Multicast Routing Table MIB-IPMROUTE-MIB. This table is augmented to cPimInetMRouteNextHopTable. This table is not supported.
cpimExtBidirDFTable	(conceptual) Table listing the Per-RP DF[1] Election state for each interface for all the RPs in Bidir mode.
cpimExtRPSetTable	(conceptual) Table listing PIM information for available RPs for IP multicast groups. An entry is learnt from one of {static, bsr, embedded} methods, as defined by the cpimExtRPSetType object. When the cpimExtRPSetType object has a value {static}, the entry is a mapping provided by user-configuration. A value of {embedded} indicates that the RP-address is embedded in the Group-address. When the value is {bsr}, this entry is obtained from received Candidate-RP-Advertisements when the local router is the BSR, and is obtained from received RP-Set messages when the local router is not the BSR.
cpimExtCRPTable	(conceptual) Table listing the IP multicast groups for which the local router is to advertise itself as a Candidate-RP when the value of cPimComponentCRPHoldTime is non-zero. If this table is empty, the local router advertises itself as a Candidate-RP for all groups (providing the value of cPimComponentCRPHoldTime is non- zero). This table is not supported.

[1] DF = designated forwarder

MIB Constraints

This table lists the constraints on objects in the CISCO-IETF-PIM-EXT-MIB.

Table 1-63 CISCO-IETF-PIM-EXT-MIB Constraints

MIB Object	Notes

Table 1-63 CISCO-IETF-PIM-EXT-MIB Constraints

MIB Object	Notes
cpimExtIfTable	Not supported
cpimExtNbrTable	Not supported
cpimExtMRouteTable	Not supported
cpimExtMRouteNextHopTable	Not supported
cpimExtCRPTable	Not supported

CISCO-IETF-PW-MIB

The CISCO-IETF-PW-MIB contains managed object definitions for pseudo wire operations. The indexes of CISCO-IETF-PW-MIB are also used to index the PSN-specific tables and the VC-specific tables. This MIB enables the use of the underlying PSN.

This table lists the tables associated with this MIB.

Table 1-64 CISCO-IETF-PW-MIB Tables and Descriptions

Name	Description
cpwVcTable	This table specifies information for connecting various emulated services to various tunnel type.
cpwVcPerfCurrentTable	This table provides per-VC performance information for the current interval.
cpwVcPerfIntervalTable	This table provides per-VC performance information for each interval.
cpwVcPerfTotalTable	This table provides per-VC Performance information from VC start time.
cpwVcIdMappingTable	This table provides reverse mapping of the existing VCs based on vc type and VC ID ordering. This table is typically useful for EMS ordered query of existing VCs.
cpwVcPeerMappingTable	This table provides reverse mapping of the existing VCs based on vc type and VC ID ordering. This table is typically useful for EMS ordered query of existing VCs.

CISCO-IETF-PW-ENET-MIB

The CISCO-IETF-PW-ENET-MIB describes a model for managing Ethernet point-to-point pseudo wire services over a Packet Switched Network (PSN).

This table lists the tables associated with this MIB.

Table 1-65 CISCO-IETF-PW-ENET-MIB Tables and Descriptions

Name	Description
cpwVcEnetTable	This table contains the index to the Ethernet tables associated with this ETH VC, the VLAN configuration and VLAN mode.
cpwVcEnetMplsPriMappingT	This table may be used for MPLS PSNs if there is a need to hold multiple VC, each with different COS, for the same user service (port + PW VLAN).

Table 1-65 CISCO-IETF-PW-ENET-MIB Tables and Descriptions

Name	**Description**
able	Such a need may arise if the MPLS network is capable of L-LSP or E-LSP without multiple COS capabilities. Each row is indexed by the cpwVcIndex and indicate the PRI bits on the packet received from the user port (or VPLS virtual port) that are classified to this VC. Note that the EXP bit value of the VC is configured in the CISCO-IETF-PW-MPLS-MIB.
cpwVcEnetStatsTable	This table contains statistical counters specific for Ethernet PW.

CISCO-IETF-PW-MPLS-MIB

The CISCO-IETF-PW-MPLS-MIB complements the CISCO-IETF-PW-MIB for pseudo wire operation over Multiprotocol Label Switching (MPLS).

This table lists the tables associated with this MIB.

Table 1-66 ***CISCO-IETF-PW-MPLS-MIB Tables and Descriptions***

Name	**Description**
cpwVcMplsTable	This table specifies information for VC to be carried over MPLS PSN.
cpwVcMplsOutboundTable	This table associates VCs using MPLS PSN with the outbound MPLS tunnels (that is toward the PSN) or the physical interface in case of VC only.
cpwVcMplsInboundTable	This table associates VCs using MPLS PSN with the inbound MPLS tunnels (that is, for packets coming from the PSN), if such association is desired (mainly for security reasons).
cpwVcMplsNonTeMappingTable	This table maps an inbound/outbound Tunnel to a VC in non-TE applications.
cpwVcMplsTeMappingTable	This table maps an inbound/outbound Tunnel to a VC in MPLS-TE applications.

CISCO-IETF-PW-TC-MIB

The CISCO-IETF-PW-TC-MIB provides textual conventions and OBJECT-IDENTITY objects to be used in pseudo wire services. This MIB has no tables.

CISCO-IETF-VPLS-BGP-EXT-MIB

The CISCO-IETF-VPLS-BGP-EXT-MIB module enables the use of any underlying Pseudo Wire network. This MIB extends the MIB module published in the RFC 4188 to manage object definitions for BGP signalled VPLS.

This table lists the tables associated with this MIB.

Table 1-67 ***CISCO-***IETF-VPLS-BGP-EXT-MIB Tables and Descriptions

Name	**Description**

Table 1-67 ***CISCO***-IETF-VPLS-BGP-EXT-MIB Tables and Descriptions

Name	Description
ciVplsBgpExtConfigTable	This table specifies information for configuring and monitoring BGP-specific parameters for VPLS. A row is automatically created when a VPLS is configured using BGP signaling. None of the read-write objects values can be changed when cvplsConfigRowStatus is in the active(1) state. Changes are allowed when the cvplsConfigRowStatus is in notInService(2) or notReady(3) states only. If the operator need to change one of the values for an active row the cvplsConfigRowStatus should be first changed to notInService(2), the objects may be changed now, and later to active(1) to re-initiate the signaling process with the new values in effect.
civplsBgpExtRTTable	This table specifies information for the list of RTs imported or exported by BGP during auto-discovery of VPLS.
ciVplsBgpExtVETable	This table associates VPLS Edge devices to a VPLS. The VEs assigned to a VPLS can be configured on a PE. This table has an expansion dependant relationship with cvplsConfigTable. For each row identified by cvplsConfigIndex, there may exist one or more rows in this table. ciVplsBgpExtVEId is the expansion index. None of the read-create objects values can be changed when ciVplsBgpExtVERowStatus is in the active(1) state. Changes are allowed when the ciVplsBgpExtVERowStatus is in notInService(2) or notReady(3) states only. If the operator need to change one of the values for an active row the ciVplsBgpExtVERowStatus should be first changed to notInService(2), the objects may be changed now, and later to active(1) to re-initiate the signaling process with the new values in effect.
ciVplsBgpExtPwBindTable	This table provides BGP-specific information for an association between a VPLS and the corresponding Pseudo Wires. A service can have more than one Pseudo Wire association. Pseudo Wires are defined in the cpwvcTable. Each row represents an association between a VPLS instance and one or more Pseudo Wires defined in the cpwVcTable in CISCO-IETF-PW-MIB. An Entry in this table in instantiated only when BGP signaling is used to configure VPLS.

CISCO-IETF-VPLS-GENERIC-MIB

The CISCO-IETF-VPLS-GENERIC-MIB contains generic managed object definitions for Virtual Private LAN services as in [L2VPN-VPLS-LDP] and [L2VPN-VPLS-BGP]. This MIB enables the use of any underlying Pseudo network.

This table lists the tables associated with this MIB.

Table 1-68 ***CISCO-IETF-VPLS-GENERIC-MIB Tables and Descriptions***

Name	Description
cvplsConfigTable	This table specifies information for configuring and monitoring VPLS.
cvplsStatusTable	This table provides information for monitoring VPLS.
cvplsPwBindTable	This table provides an association between a VPLS service and the corresponding Pseudo Wires. A service can have more than one Pseudo Wire association. Pseudo Wires are defined in the pwTable.

CISCO-IETF-VPLS-LDP-MIB

The CISCO-IETF-VPLS-LDP-MIB contains managed object definitions for LDP signaled Virtual Private LAN services as in [L2VPN-VPLS-LDP] and enables the use of any underlying Pseudo network.

This table lists the tables associated with this MIB.

Table 1-69 CISCO-IETF-VPLS-LDP-MIB Tables and Descriptions

Name	Description
cvplsLdpConfigTable	This table specifies information for configuring and monitoring LDP specific parameters for VPLS[1].
cvplsLdpPwBindTable	This table provides LDP specific information for an association between a VPLS service and the corresponding Pseudo Wires. A service can have more than one Pseudo Wire association. Pseudo Wires are defined in the cpwTable.

[1] VPLS = virtual private LAN services

CISCO-IF-EXTENSION-MIB

The CISCO-IF-EXTENSION-MIB extends the IF-MIB(RFC 2863) to add objects which provide additional information about information not available on other MIBs. This MIB replaces the OLD-CISCO-INTERFACES-MIB.

This table lists the tables associated with this MIB.

Table 1-70 CISCO-IF-EXTENSION-MIB Tables and Descriptions

Name	Description
cieIFPacketStatsTable	This table contains interface packet statistics which are not available in IF-MIB (RFC 2863).
cieIfInterfaceTable	This table contains objects which provide more information about interface properties not available in IF-MIB (RFC 2863). Some objects defined in this table may be applicable to physical interfaces only. As a result, this table may be sparse for logical interfaces.
cieIfStatusListTable	This table contains objects for providing the 'ifIndex', interface operational mode and interface operational cause for all the interfaces in the modules. This table contains one entry for each 64 interfaces in an module. This table provides efficient way of encoding 'ifIndex', interface operational mode and interface operational cause, from the point of retrieval, by combining the values a set of 64 interfaces in a single MIB object.
cieIfDot1qCustomEtherTypeTable	List of the interfaces that support the 802.1q custom Ethertype feature.
cieIfUtilTable	This table contains the interface utilization rates for inbound and outbound traffic on an interface.
cieIfDot1dBaseMappingTable	This table contains the mappings of the ifIndex of an interface to its corresponding dot1dBasePort value.
cieIfNameMappingTable	This table contains objects for providing the 'ifName' to 'ifIndex' mapping. This table contains one entry for each valid 'ifName' available in the system. Upon the first request, the implementation of this table gets all the available ifNames, and it populates the entries in this table, it maintains

Table 1-70 CISCO-IF-EXTENSION-MIB Tables and Descriptions

Name	Description
	this ifNames in a cache for ~30 seconds.

MIB Constraints

This table lists the constraints on objects in the CISCO-IF-EXTENSION-MIB.

Table 1-71 CISCO-IF-EXTENSION-MIB Constraints

MIB Object	Notes
cieSystemMtu	Not supported
cieIfDot1qCustomAdminEtherType	Not supported
cieLinkUpDownEnable	Not supported
cieStandardLinkUpDownVarbinds	Not supported
cieIfDhcpMode	Not supported
cieIfMtu	Not supported
cieIfAutoNegotiate	Not supported
cieIfKeepAliveEnabled	Not supported
cieIfLastInTime	Not supported
cieIfLastOutTime	Not supported
cieIfLastOutHangTime	Not supported
cieIfPacketDiscontinuityTime	Not supported
cieIfStatusListIndex	Not supported
cieInterfaceOwnershipBitmap	Not supported
cieIfInterfaceDiscontinuityTime	Not supported
CieIfVlStatsEntry	Not supported
cieIfNoDropVlInPkts	Not supported
cieIfNoDropVlInOctets	Not supported
cieIfNoDropVlOutPkts	Not supported
cieIfNoDropVlOutOctets	Not supported
cieIfDropVlInPkts	Not supported
cieIfDropVlInOctets	Not supported
cieIfDropVlOutPkts	Not supported
MIB Object	**Notes**

Table 1-71 ***CISCO-IF-EXTENSION-MIB Constraints***

MIB Object	Notes
cieIfDropVlOutOctets	Not supported
cieIfIndexPersistence	Not supported
cieIfIndexPersistenceEnabled	Not supported
cieIfIndexPersistenceControl	Not supported
cieDelayedLinkUpDownNotifEnable	Not supported
cieDelayedLinkUpDownNotifDelay	Not supported
cieIfIndexGlobalPersistence	Not supported
cieLinkUpDownConfig	Not supported
cieIfSpeedReceive	Not supported
cieIfHighSpeedReceive	Not supported
cieIfOwner	Not supported
cieIfSharedConfig	Not supported
cieIfSpeedGroupConfig	Not supported
cieIfTransceiverFrequencyConfig	Not supported
cieIfFillPatternConfig	Not supported
cieIfIgnoreBitErrorsConfig	Not supported
cieIfIgnoreInterruptThresholdConfig	Not supported

Note Caching of all tables is not supported.

CISCO-IP-STAT-MIB

The CISCO-IP-STAT-MIB incorporates objects to provide support for the Cisco IP statistics as implemented in command interfaces.

MIB Tables

This table lists the tables in CISCO-IP-STAT-MIB:

Table 1-72 ***CISCO-IP-STAT-MIB Tables***

Name	Description
cipPrecedenceTable	A table of entries sorted by the precedence of IP packets. The table is created and deleted via ip accounting command line interface.
cipMacTable	A table is created and deleted via ip accounting command line

Table 1-72 CISCO-IP-STAT-MIB Tables

Name	Description
	interface.
cipMacFreeTable	A table of free space available to store new MAC address information.
cipPrecedenceXTable	This table contains additional objects for the cipPrecedenceTable.
cipMacXTable	This table contains additional objects for the cipMacTable.

CISCO-MEMORY-POOL-MIB

The CISCO-MEMORY-POOL-MIB contains objects that represents the different types of memory pools that may be present in a managed device. Memory pools are categorized into two groups:

- Predefined pools
- Dynamic pools

This table lists the tables associated with this MIB.

Table 1-73 CISCO-MEMORY-POOL-MIB Tables and Descriptions

Name	Description
ciscoMemoryPoolTable	Table of memory pool monitoring entries.
ciscoMemoryPoolUtilizationTable	Table of memory pool utilization entries. Each of the objects provides a general idea of how much of the memory pool has been used over a given period of time. It is determined as a weighted decaying average.

CISCO-NTP-MIB

The CISCO-NTP-MIB provides mechanisms to monitor an Network Time Protocol (NTP) server. The (NTP) Version 3 is used to synchronize timekeeping among a set of distributed time servers and clients. The service model is based on a returnable-time design which depends only on measured clock offsets, but does not require reliable message delivery. The synchronization subnet uses a self-organizing, hierarchical master-slave configuration, with synchronization paths determined by a minimum-weight spanning tree. While multiple masters (primary servers) may exist, there is no requirement for an election protocol.

In the NTP model several primary reference sources, synchronized by wire or radio to national standards, are connected to widely accessible resources, such as backbone gateways, and operated as primary time servers. The purpose of NTP is to convey timekeeping information from these servers to other time servers via the Internet and also to cross-check clocks and mitigate errors because of equipment or propagation failures. Some number of local-net hosts or gateways, acting as secondary time servers, run NTP with one or more of the primary servers. To reduce the protocol overhead, the secondary servers distribute time via NTP to the remaining local-net hosts. In the interest of reliability, selected hosts can be equipped with less accurate but less expensive radio clocks and used for backup in case of failure of the primary or secondary servers or communication paths between them.

NTP is designed to produce three products: clock offset, round-trip delay, and dispersion, all of which are relative to a selected reference clock. Clock offset represents the amount to adjust the local clock to bring it into correspondence with the reference clock. Roundtrip delay provides the capability to launch a message to arrive at the reference clock at a specified time. Dispersion represents the maximum error of the local clock relative to the reference clock. Because most host time servers synchronize via another peer time server, there are two components in each of these three products, those determined by the peer relative to the primary reference source of standard time and those measured by the host relative to the peer. Each of these components are maintained separately in the protocol to facilitate error control and management of the subnet itself. They provide not only precision measurements of offset and delay, but also definitive maximum error bounds, so that the user interface can determine not only the time, but the quality of the time as well.

In what may be the most common client/server model, a client sends an NTP message to one or more servers and processes the replies as received. The server interchanges addresses and ports, overwrites certain fields in the message, recalculates the checksum and returns the message immediately.

Information included in the NTP message allows the client to determine the server time with respect to local time and adjust the local clock accordingly. Also, the message includes information to calculate expected timekeeping accuracy and reliability, as well as select the best from possibly several servers. Although the client/server model may suffice for use on local nets involving a public server and perhaps many workstation clients, the full generality of NTP requires distributed participation of a number of client/servers or peers arranged in a dynamically reconfigurable, hierarchically distributed configuration. It also requires sophisticated algorithms for association management, data manipulation and local-clock control.

This table lists the tables associated with this MIB.

Table 1-74 *CISCO-NTP-MIB Tables and Descriptions*

Name	Description
cntpPeersVarTable	This table provides information on the peers with which the local NTP server has associations. The peers are also NTP servers but running on different hosts.
cntpFilterRegisterTable	Contains NTP state variables used by the NTP clock filter and selection algorithms. This table depicts a shift register. Each stage in the shift register is a 3-tuple consisting of the measured clock offset, measured clock delay, and measured clock dispersion associated with a single observation. An important factor affecting the accuracy and reliability of time distribution is the complex of algorithms used to reduce the effect of statistical errors and falsetickers because of failure of various subnet components, reference sources or propagation media. The NTP clock-filter and selection algorithms are designed to do exactly this. The objects in the filter register table below are used by these algorithms to minimize the error in the calculated time.

MIB Constraints

Note CISCO-NTP-MIB has very limited support. cntpSysSrvStatus is supported.

This table lists the constraints on objects in the CISCO-NTP-MIB.

Table 1-75 ***CISCO***-NTP-MIB Constraints

MIB Object	Notes
cntpPeersVarTable	Not supported
cntpFilterRegisterTable	Not supported

CISCO-OTN-IF-MIB

The CISCO-OTN-IF-MIB defines the managed objects for physical layer characteristics of DWDM optical channel interfaces and performance statistics objects for protocol specific error counters in DWDM optical devices.

Performance monitoring (PM) parameters are used by service providers to gather, store, set thresholds for and report performance data for early detection of problems. Thresholds are used to set error levels for each PM parameter. During the accumulation cycle, if the current value of a performance monitoring parameter reaches or exceeds its corresponding threshold value, a threshold crossing alarm (TCA) is generated. The TCAs provide early detection of performance degradation.

This table lists the tables associated with this MIB.

Table 1-76 ***CISCO***-OTN-IF-MIB Tables and Descriptions

Name	Description
coiIfControllerTable	This table provides management information for physical layer related attributes of interfaces with an ifType of opticalChannel (195).
coiOtnNearEndThresholdsTable	This table provides objects for configuring OTN (G.709) near end error thresholds on interfaces of ifType opticalChannel (195).
coiOtnFarEndThresholdsTable	This table provides objects for configuring OTN (G.709) thresholds for far end of interfaces of ifType opticalChannel (195).
coiOtnNearEndCurrentTable	This table contains the cumulative OTN (G.709) PM statistics for the near end of interfaces of ifType opticalChannel (195). The statistics are for the current interval of interval type identified by coiOtnNearEndCurIntervalType. The current PM statistics is the accumulated statistics for the time period defined by the interval type.
coiOtnFarEndCurrentTable	This table contains the cumulative OTN (G.709) PM stats for the far end of interfaces of ifType opticalChannel (195). The statistics are for the current interval of interval type identified by coiOtnFarEndCurIntervalType. The current PM statistics is the accumulated statistics for the time period defined by the interval type.
coiOtnNearEndIntervalTable	This table contains historical cumulative OTN (G.709) PM stats for the near end of interfaces of ifType opticalChannel (195), for the interval type identified by the index coiOtnNearEndIntervalType and the interval number as identified by the index coiOtnNearEndIntervalNum. The PM statistics is the accumulated stats for the time period defined by the interval type in the time interval as defined by interval number.
coiOtnFarEndIntervalTable	This table contains historical cumulative OTN (G.709) PM stats for the far end interfaces of ifType opticalChannel (195), for the interval type identified by the index coiOtnFarEndIntervalType and the interval number as identified by coiOtnFarEndIntervalNum. The PM statistics is the

Table 1-76 ***CISCO***-OTN-IF-MIB Tables and Descriptions

Name	Description
	accumulated stats for the time period defined by the interval type in the time interval as defined by interval number.
coiFECThresholdsTable	This table contains the configurable thresholds for Forward Error Correction statistics.
coiFECCurrentTable	This table contains the cumulative FEC PM stats for the interfaces of ifType opticalChannel (195) for the current interval of interval type identified coiFECCurIntervalType.
coiFECIntervalTable	This table contains historical cumulative FEC PM stats for the interfaces of ifType opticalChannel (195), for the interval type identified by the index coiFECIntervalType and the interval number as identified by index coiFECIntervalNum. The PM statistics is the accumulated stats for the time period defined by the interval type in the time interval as defined by interval number.

CISCO-PIM-MIB

The CISCO-PIM-MIB defines the Cisco specific variables for Protocol Independent Multicast (PIM) management. These definitions are an extension of those defined in the UETF PIM MIB (RFC 2934). This MIB has no tables. A Management Station pinging different Network elements can use this MIB to ping and get back the results if the Network Element is accessible or not. The number of packets, packet size, timeout, delay can be set to the appropriate values and tested. This MIB is superseded by the CISCO-RTTMON-MIB that provides this functionality in addition to other features.

CISCO-PING-MIB

The CISCO-PING-MIB is used to determine connectivity and reachability of network elements and devices via use of the PING protocol.

This table lists the tables associated with this MIB.

Table 1-77 ***CISCO-PING-MIB Tables and Descriptions***

Name	Description
ciscoPingTable	Ping request entry. A management station wishing to create an entry should first generate a pseudo-random serial number to be used as the index to this sparse table. The station should then create the associated instance of the row status and row owner objects. It must also, either in the same or in successive PDUs, create the associated instance of the protocol and address objects. It should also modify the default values for the other configuration objects if the defaults are not appropriate. After the appropriate instance of all the configuration objects have been created, either by an explicit SNMP set request or by default, the row status should be set to active to initiate the request. Note that this entire procedure may be initiated via a single set request which specifies a row status of createAndGo as well as specifies valid values for the non-defaulted configuration objects. After the ping sequence has been activated, it cannot be stopped—it runs until the configured number of packets have been sent. After the sequence completes, the management station should retrieve the values of the status objects of interest, and should then delete the entry. To

Table 1-77 ***CISCO-PING-MIB Tables and Descriptions***

Name	Description
	prevent old entries from clogging the table, entries are aged out, but an entry is never deleted within 5 minutes of completing barring an explicit delete request from the management station.

CISCO-PROCESS-MIB

The CISCO-PROCESS-MIB describes active system processes. Virtual Machine refers to those OS which can run the code or process of a different executional model OS. Virtual processes assume the executional model of a OS which is different from Native IOS. Virtual Processes are also referred to as Tasks. Thread is a sequence of instructions to be executed within a program. A thread which adheres to POSIX standard is referred to as a POSIX thread.

This table lists the tables associated with this MIB.

Table 1-78 ***CISCO-PROCESS-MIB Tables and Descriptions***

Name	Description
cpmCPUTotalTable	Table of overall CPU statistics.
cpmProcessTable	Table of generic information on all active processes on this device.
cpmProcessExtTable	This table contains information that may or may not be available on all cisco devices. It contains additional objects for the more general cpmProcessTable. This object is deprecated by cpmProcessExtRevTable.
cpmProcessExtRevTable	This table contains information that may or may not be available on all Cisco devices. It contains additional objects for the more general cpmProcessTable. This object deprecates cpmProcessExtTable.
cpmCPUThresholdTable	This table contains the information about the thresholding values for CPU, configured by the user.
cpmCPUHistoryTable	List of CPU utilization history entries.
cpmThreadTable	This table contains generic information about POSIX threads in the device.
cpmVirtualProcessTable	This table contains information about virtual processes in a virtual machine.
cpmCPUProcessHistoryTable	List of process history entries. This table contains CPU utilization of processes which crossed the cpmCPUHistoryThreshold.

CISCO-RF-MIB

The CISCO-RF-MIB provides configuration control and status for the Redundancy Framework (RF) subsystem. RF provides a mechanism for logical redundancy of software functionality and is designed to support 1:1 redundancy on Route Switch Processors (RSPs). Redundancy duplicates data elements and software functions to provide an alternative in case of failure.

Note For information about the levels of redundancy, see Appendix A, "Using MIBs."

This table lists the tables associated with this MIB.

Table 1-79 ***CISCO-RF-MIB Tables and Descriptions***

Name	Description
cRFStatusRFModeCapsTable	This table containing a list of redundancy modes that can be supported on the device.
cRFStatusRFClientTable	This table contains a list of RF clients that are registered on the device. RF clients are applications that have registered with the RF to receive RF events and notifications. The purpose of RF clients is to synchronize any relevant data with the standby unit.
cRFHistorySwitchOverTable	Table that tracks the history of all switchovers that have occurred since system initialization. The maximum number of entries permissible in this table is defined by cRFHistoryTableMaxLength. When the number of entries in the table reaches the maximum limit, the next entry would replace the oldest existing entry in the table.

MIB Constraints

This table lists the constraints on objects in the CISCO-RF-MIB.

Table 1-80 CISCO-RF-MIB Constraints

MIB Object	Notes
cRFCfgGroup	
cRFCfgSplitMode	Object is deprecated.
cRFCfgRedundancyMode	Values: 6, 7, and 8.
cRFCfgMaintenanceMode	Read-only. Supported value is false (2).
cRFHistoryGroup	
cRFHistory	There are three switchover modes: coldstandby, warmstandby, and hoststandby. The only entries saved are those generated from a hot standby switchover.

Note SNMP process placement was introduced in Cisco IOS XR Release 3.8.3. cRFStatusRFClientTable in CISCO-RF-MIB lists the status of all processes on DSC and their redundancy status. However, the redundancy status of all the processes (for example bgp, ospf) that are placeable is not correct when the process is placed on a different RP or DRP. To overcome this issue, use RFClientStatus definition to get redundancy information about the process and to get the process state use Processmib.

CISCO-RTTMON-MIB

The CISCO-RTTMON-MIB defines a MIB for Round Trip Time (RTT) monitoring of a list of targets, using a variety of protocols.

This table lists the tables associated with this MIB.

Table 1-81 CISCO-RTTMON-MIB Tables and Descriptions

MIB Object	Notes
cRFCfgGroup	
cRFCfgSplitMode	Object is deprecated.
cRFCfgRedundancyMode	Values: 6, 7, and 8.
cRFCfgMaintenanceMode	Read-only. Supported value is false (2).

Table 1-81 CISCO-RTTMON-MIB Tables and Descriptions

MIB Object	Notes
cRFHistoryGroup	
cRFHistory	There are three switchover modes: coldstandby, warmstandby, and hoststandby. The only entries saved are those generated from a hot standby switchover.
rttMonApplSupportedRttTypesTable	Table of which contains the supported Rtt Monitor Types. See the RttMonRttType textual convention for the definition of each type.
rttMonApplSupportedProtocolsTable	Table of which contains the supported Rtt Monitor Protocols. See the RttMonProtocol textual convention for the definition of each protocol.
rttMonApplPreConfigedTable	Not supported.
rttMonApplAuthTable	Not supported.
rttMonCtrlAdminTable	Table of RTT monitoring definitions. The RTT administration control is in multiple tables. This first table, is used to create a conceptual RTT control row. The following tables contain objects which configure scheduling, information gathering, and notification/ trigger generation. All of these tables create the same conceptual RTT control row as this table using this table index as their own index. This table is limited in size by the agent implementation. The object rttMonApplNumCtrlAdminEntry reflects this tables maximum number of entries.
rttMonEchoAdminTable	Table that contains RTT specific definitions. This table is controlled via the rttMonCtrlAdminTable. Entries in this table are created via the rttMonCtrlAdminStatus object.
rttMonFileIOAdminTable	Not supported.
rttMonScriptAdminTable	Not supported.
rttMonScheduleAdminTable	Table of RTT monitoring scheduling specific definitions. This table is controlled via the rttMonCtrlAdminTable. Entries in this table are created via the rttMonCtrlAdminStatus object.
rttMonReactAdminTable	Not supported. This table was replaced by rttMonReactTable.
rttMonStatisticsAdminTable	Table of Round Trip Time (RTT) monitoring statistics definitions. The definitions in this table control what and how many entries are placed into the rttMonStatsCaptureTable. The statistics capture table is a rollover table. When the rttMonStatisticsAdminNumHourGroups index value exceeds its value defined in this table, the oldest corresponding group is deleted and is replaced with the new group. All other indices only fill to there maximum size. **Note**: The maximum size of this table is defined to be the product of the ttMonCtrlAdminIndex times rttMonStatisticsAdminNumHourGroups times rttMonStatisticsAdminNumPaths times rttMonStatisticsAdminNumHops times rttMonStatisticsAdminNumDistBuckets. **Note**: Each of the 'Num' objects values in this have a special behavior. When one of the objects is set to a value larger than the RTT application can support the set succeeds, but the resultant value is set to the applications maximum value. The setting management station must reread this object to verify the actual value. This table augments the

Table 1-81 CISCO-RTTMON-MIB Tables and Descriptions

MIB Object	Notes
	rttMonCtrlAdminTable
rttMonHistoryAdminTable	Table of RTT monitoring history definitions. The definitions in this table control what and how many entries are placed into the ttMonHistoryCollectionTable. The history collection table is a rollover table. When the rttMonHistoryAdminNumLives index value exceeds its value defined in this table, the oldest corresponding 'lives' group are deleted and are replaced with the new 'livesgroup. All other indices only fill to their maximum size. **Note:** The maximum size of this table is defined to be the product of the rttMonCtrlAdminIndex times rttMonHistoryAdminNumLives times rttMonHistoryAdminNumBuckets times rttMonHistoryAdminNumSamples. **Note:** Each of the 'Num' objects values in this have a special behavior. When one of the objects is set to a value larger than the RTT application can support the set succeeds, but the resultant value is set to the applications maximum value. The setting management station must reread this object to verify the actual value. **Note:** This table is not applicable to http and jitter probes.
rttMonCtrlOperTable	Table that contains the Operational values for the probe, and the conceptual RTT control row. This table augments the rttMonCtrlAdminTable.
rttMonLatestRttOperTable	Table that contains the status of latest RTT operation. When the RttMonRttType is 'pathEcho', operations performed to the hops along the path will be recorded inthis table. This table augments the RTT definition table, rttMonCtrlAdminTable
rttMonLatestHTTPOperTable	Not supported.
rttMonLatestJitterOperTable	Table that contains the status of the latest Jitter operation.
rttMonReactTriggerAdminTable	Table that contains the list of conceptual RTT control rows that start to collect data when a reaction condition is violated and when rttMonReactAdminActionType is set to one of the following: ■ triggerOnly ■ trapAndTrigger ■ nmvtAndTrigger ■ trapNmvtAndTrigger or when a reaction condition is violated and when any of the row in rttMonReactTable has rttMonReactActionType as one of the following: ■ triggerOnly ■ trapAndTrigger The goal of this table is to define one or more additional conceptual RTT control rows that become active and start to collect additional history and statistics (depending on the rows configuration values), when a problem has been detected. If the conceptual RTT control row is undefined, and a trigger occurs, no action takes place. If the conceptual RTT control row is

Table 1-81 CISCO-RTTMON-MIB Tables and Descriptions

MIB Object	Notes
	scheduled to start at a later time, triggering that row has no effect. If the conceptual RTT control row is currently active, triggering that row has no effect on that row, but the rttMonReactTriggerOperState object transitions to 'active'. An entry in this table can only be triggered when it is not currently in a triggered state. The object rttMonReactTriggerOperState reflects the state of each entry in this table.
rttMonReactTriggerOperTable	Table of which contains the operational state of each entry in the rttMonReactTriggerAdminTable. This table augments the RTT trigger definition table, rttMonReactTriggerAdminTable.
rttMonEchoPathAdminTable	Table to store the hop addresses in a Loose Source Routing path. Response times are computed along the specified path using ping. This maximum table size is limited by the size of the maximum number of hop addresses that can fit in an IP header, which are eight. The object rttMonEchoPathAdminEntry reflects this tables maximum number of entries. This table is coupled with rttMonCtrlAdminStatus.
rttMonGrpScheduleAdminTable	Not supported
rttMplsVpnMonCtrlTable	Table of Auto SAA Layer 3 MPLS VPN definitions. The Auto SAA Layer 3 MPLS VPN administration control is in multiple tables. This first table is used to create a conceptual Auto SAA Layer 3 MPLS VPN control row. The following tables contain objects which used in type specific configurations, scheduling and reaction configurations. All of these tables create the same conceptual control row as this table using this table index as their own index. In order for a row in this table to become active, the following objects must be defined. rttMplsVpnMonCtrlRttType, rttMplsVpnMonCtrlVrfName, and rttMplsVpnMonSchedulePeriod.
rttMplsVpnMonTypeTable	Table that contains Auto SAA Layer 3 MPLS VPN configured RTT operation specific definitions. Table is controlled via the rttMplsVpnMonCtrlTable. Entries in this table are created via the rttMplsVpnMonCtrlStatus object
rttMplsVpnMonScheduleTable	Table of Auto SAA Layer 3 MPLS VPN monitoring scheduling specific definitions. This table is controlled via the rttMplsVpnMonCtrlTable. Entries in this table are created via the rttMplsVpnMonCtrlStatus object.
rttMplsVpnMonReactTable	Table of Auto SAA Layer 3 MPLS VPN Notification definitions. This table augments the rttMplsVpnMonCtrlTable.
rttMonReactTable	Table that contains the reaction configurations. Each conceptual row in rttMonReactTable corresponds to a reaction configured for the probe defined in rttMonCtrlAdminTable. For each reaction configured for a probe there is an entry in the table. Each Probe can have multiple reactions and hence there can be multiple rows for a particular probe. This table is coupled with rttMonCtrlAdminTable.
rttMonStatsCaptureTable	The statistics capture database. The statistics capture table contains summarized information of the results for a conceptual RTT control row. A rolling accumulated history of this information is maintained in a series of hourly 'group(s)'. Each 'group' contains a series of 'path(s)', each 'path' contains a series of 'hop(s)', and each 'hop' contains a series of

Table 1-81 CISCO-RTTMON-MIB Tables and Descriptions

MIB Object	Notes
	'statistics distribution bucket'. Each conceptual statistics row has a current hourly group, into which RTT results are accumulated. At the end of each hour a new hourly group is created which then becomes current. The counters and accumulators in the new group are initialized to zero. The previous group is kept in the table until the table contains rttMonStatisticsAdminNumHourGroups groups for the conceptual statistics row; at this point, the oldest group is discarded and is replaced by the newly created one. The hourly group is uniquely identified by the rttMonStatsCaptureStartTimeIndex object. If the activity for a conceptual RTT control row ceases because the rttMonCtrlOperState object transitions to 'inactive', the corresponding current hourly group in this table is 'frozen', and a new hourly group is created when activity is resumed. If the activity for a conceptual RTT control row ceases because the rttMonCtrlOperState object transitions to 'pending' this whole table will be cleared and reset to its initial state. When the RttMonRttType is 'pathEcho', the path exploration RTT request statistics will not be accumulated in this table. **Note:** When the RttMonRttType is 'pathEcho', a source to target rttMonStatsCapturePathIndex path will be created for each rttMonStatsCaptureStartTimeIndex to hold all errors that occur when a specific path had not been found or connection has not be setup. Using this rttMonStatsCaptureTable, a managing application can retrieve summarized data from accurately measured periods, which is synchronized across multiple conceptual RTT control rows. With the new hourly group creation being performed on a 60-minute period, the managing station has plenty of time to collect the data, and need not be concerned with the vagaries of network delays and lost PDU's when trying to get matching data. Also, the managing station can spread the data gathering over a longer period, which removes the need for a flood of get requests in a short period which otherwise would occur.
rttMonStatsCollectTable	Not supported.
rttMonStatsTotalsTable	Not supported.
rttMonHTTPStatsTable	Not supported.
rttMonJitterStatsTable	Jitter statistics collection database. The Jitter statistics table contains summarized information of the results for a conceptual RTT control row. A rolling accumulated history of this information is maintained in a series of hourly 'group(s)'. The operation of this table is same as that of rttMonStatsCaptureTable, except that this table stores 2 hours of data.
rttMonLpdGrpStatsTable	Auto SAA Layer 3 MPLS VPN LPD Group Database. The LPD Group statistics table contains summarized performance statistics for the LPD group. LPD Group–Set of 'single probes' which are subset of the 'lspGroup' probe traversing set of paths between two PE end points are grouped together and called as the LPD group. The LPD group is uniquely referenced by the LPD Group ID. A rolling accumulated history of this information is maintained in a series of hourly 'group(s)'. Each conceptual statistics row has a current hourly group, into which RTT results are accumulated. At the end of each hour a new hourly group is

Table 1-81 CISCO-RTTMON-MIB Tables and Descriptions

MIB Object	Notes
	created which then becomes current. The counters and accumulators in the new group are initialized to zero. The previous group(s) is kept in the table until the table contains rttMplsVpnMonTypeLpdStatHours groups for the conceptual statistics row; at this point, the oldest group is discarded and is replaced by the newly created one. The hourly group is uniquely identified by the rttMonLpdGrpStatsStartTimeIndex object.
rttMonHistoryCollectionTable	History collection database. The history table contains a point by point rolling history of the most recent RTT operations for each conceptual RTT control row. The rolling history of this information is maintained in a series of 'live(s)', each containing a series of 'bucket(s)', each 'bucket' contains a series of 'sample(s)'. Each conceptual history row can have lives. A life is defined by the rttMonCtrlOperRttLife object. A new life is created when rttMonCtrlOperState transitions 'active'. When the number of lives becomes greater than rttMonHistoryAdminNumLives the oldest life is discarded and a new life is created by incrementing the index. The path exploration RTT operation is kept as an entry in this table.

MIB Constraints

This table lists the constraints on objects in the CISCO-RTTMON-MIB.

Table 1-82 CISCO-RTTMON-MIB Constraints

MIB Object	Notes
rttMonApplPreConfigedTable	Not supported–No back end IP SLA.
rttMonApplAuthTable	Not supported–No back end IP SLA.
rttMonFileIOAdminTable	Not supported–No back end IP SLA.
rttMonScriptAdminTable	Not supported–No back end IP SLA.
rttMonReactAdminTable	Not supported. This table is replaced by rttMonReactTable.
rttMonLatestHTTPOperTable	Not supported–IP SLA in XR does not support HTTP probes.
rttMonGrpScheduleAdminTable	Not supported–No back end IP SLA.
rttMonStatsCollectTable	Not supported–No back end IP SLA.
rttMonStatsTotalsTable	Not supported–No back end IP SLA.
rttMonHTTPStatsTable	Not supported–IP SLA in XR does not support HTTP probes.

CISCO-SONET-MIB

The CISCO-SONET-MIB describes SONET/SDH interfaces objects. This is an extension to the standard SONET MIB (RFC 2558).

This table lists the tables associated with this MIB.

Table 1-83 CISCO-SONET-MIB Tables and Descriptions

Name	Description
csConfigTable	SONET/SDH configuration table. This table has objects for configuring sonet lines.

Table 1-83 CISCO-SONET-MIB Tables and Descriptions

Name	Description
csVTConfigTable	This table contains objects to configure the VT/VC[1] related properties of SONET/SDH lines.
csApsConfigTable	This table contains objects to configure APS[2] feature in a SONET Line. APS is the ability to configure a pair of SONET lines for redundancy so that the hardware automatically switches the active line from working line to the protection line or vice versa, within 60 ms, when the active line fails.
cssTotalTable	SONET/SDH Section Total table. It contains the cumulative sum of the various statistics for the 24 hour period preceding the current interval. The object 'sonetMediumValidIntervals' from RFC 2558 contains the number of 15-minute intervals that have elapsed since the line is enabled.
cssTraceTable	SONET/SDH Section Trace table. This table contains objects for tracing the sonet section.
cslTotalTable	SONET/SDH Line Total table. It contains the cumulative sum of the various statistics for the 24-hour period preceding the current interval. The object 'sonetMediumValidIntervals' from RFC 2558 contains the number of 15-minute intervals that have elapsed since the line is enabled.
cslFarEndTotalTable	SONET/SDH Far End Line Total table. It contains the cumulative sum of the various statistics for the 24-hour period preceding the current interval. The object 'sonetMediumValidIntervals' from RFC 2558 contains the number of 15-minute intervals that have elapsed since the line is enabled.
cspTotalTable	SONET/SDH Path Total table. It contains the cumulative sum of the various statistics for the 24-hour period preceding the current interval.The object 'sonetMediumValidIntervals' from RFC 2558 contains the number of 15-minute intervals that have elapsed since the line is enabled.
cspFarEndTotalTable	SONET/SDH Far End Path Total table. Far End is the remote end of the line. The table contains the cumulative sum of the various statistics for the 24-hour period preceding the current interval. The object 'sonetMediumValidIntervals' from RFC 2558 contains the number of 15-minute intervals that have elapsed since the line is enabled.
cspTraceTable	SONET/SDH Path Trace table. This table contains objects for tracing the sonet path.
csStatsTable	SONET/SDH Section statistics table. This table maintains the number of times the line encountered LOS, LOF, AISs, RFIs.
cspConfigTable	Entry in Cisco extension to the SONET path current table." Augments sonetPathCurrentEntry.
csAu4Tug3ConfigTable	This table contains objects to configure the VC[3] related properties of a TUG-3 within a AU-4 paths.

[1] VT/VC = Virtual Tributary/Virtual Container

[2] APS = automatic protection switching

[3] VC = Virtual Container

CISCO-SYSLOG-MIB

The CISCO-SYSLOG-MIB contains objects to manage all the system log messages generated by the Cisco IOS XR Software. The MIB provides a way to access the syslog messages through SNMP. All Cisco IOS XR syslog messages contain the message name and its severity, message text, the name of the entity generating the message, and an optional time stamp. The MIB also contains a history of syslog messages and counts related to syslog messages.

Note The MIB does not keep track of messages generated from debug commands entered through the CLI.

This table lists the tables associated with this MIB.

Table 1-84 ***CISCO-SYSLOG-MIB Tables and Descriptions***

Name	**Description**
clogHistoryTable	Table of syslog messages generated by this device. All 'interesting' syslog messages (that is, severity <=clogMaxSeverity) are entered into this table.
clogServerConfigTable	This table contains entries that allow application to configure syslog servers for the system. The maximum number of entries that can be created for this table is limited by the object clogMaxServers.

MIB Constraints

This table lists the constraints on objects in the CISCO-SYSLOG-MIB.

Table 1-85 ***CISCO-SYSLOG-MIB Constraints***

MIB Object	**Notes**
clogServerMaxTable	Not supported

CISCO-SYSTEM-MIB

The CISCO-SYSTEM-MIB provides a standard set of basic system information. This MIB module contains Cisco-defined extensions to the systemGroup. This MIB has no tables.

CISCO-TCP-MIB

The CISCO-TCP-MIB is an extension to the IETF MIB module for managing TCP implementations.

This table lists the tables associated with this MIB.

Table 1-86 ***CISCO-TCP-MIB Tables and Descriptions***

Name	**Description**
ciscoTcpConnTable	Table containing TCP connection-specific information.

CISCO-VLAN-IFTABLE-RELATIONSHIP-MIB

The CISCO-VLAN-IFTABLE-RELATIONSHIP-MIB lists VLAN-id and ifIndex information for routed VLAN interfaces. The MIB contains entries for all sub-interfaces that have a basic 802.1Q VLAN Id configured, but excludes any sub-interfaces configured with a more complex encapsulation (that is double tagged, 802.1ad tagged, VLAN ranges).

This table lists the tables associated with this MIB.

Table 1-87 CISCO-VLAN-IFTABLE-RELATIONSHIP-MIB Tables and Descriptions

Name	**Description**
cviVlanInterfaceIndexTable	cviVlanInterfaceIndexTable provides a way to translate a VLAN-id in to an ifIndex, so that the routed VLAN interface routing configuration can be obtained from interface entry in ipRouteTable. Note that some routers can have interfaces to multiple VLAN management domains, and therefore can have multiple routed VLAN interfaces which connect to different VLANs having the same VLAN-id. Thus, it is possible to have multiple rows in this table for the same VLAN-id. The cviVlanInterfaceIndexTable also provides a way to find the VLAN-id from an ifTable VLAN ifIndex.

DS1-MIB

The DS1-MIB module describes DS1, E1, DS2, and E2 interface objects.

This table lists the tables associated with this MIB.

Table 1-88 DS1-MIB Tables and Descriptions

Name	**Description**
dsx1ConfigTable	DS1 Configuration table.
dsx1CurrentTable	DS1 current table contains various statistics being collected for the current 15-minute interval.
dsx1IntervalTable	DS1 Interval Table contains various statistics collected by each DS1 Interface over the previous 24-hours of operation. The past 24 hours are broken into 96 completed 15-minute intervals. Each row in this table represents one such interval (identified by dsx1IntervalNumber) for one specific instance (identified by dsx1IntervalIndex).
dsx1TotalTable	DS1 Total Table contains the cumulative sum of the various statistics for the 24-hour period preceding the current interval.
dsx1ChanMappingTable	DS1 Channel Mapping table. This table maps a DS1 channel number on a particular DS3 into an ifIndex. In the presence of DS2s, this table can be used to map a DS2 channel number on a DS3 into an ifIndex, or used to map a DS1 channel number on a DS2 onto an ifIndex.
dsx1FarEndCurrentTable	DS1 Far End Current table contains various statistics being collected for the current 15-minute interval. The statistics are collected from the far end messages on the Facilities Data Link. The definitions are the same as described for the near-end information.
dsx1FarEndIntervalTable	DS1 Far End Interval Table contains various statistics collected by each DS1 interface over the previous 24-hours of operation. The past 24 hours are broken into 96 completed 15-minute intervals. Each row in this table represents one such interval (identified by dsx1FarEndIntervalNumber) for one specific instance (identified by

Table 1-88 DS1-MIB Tables and Descriptions

Name	Description
	dsx1FarEndIntervalIndex).
dsx1FarEndTotalTable	DS1 Far End Total Table contains the cumulative sum of the various statistics for the 24-hour period preceding the current interval.

dsx1FracTable	Table is deprecated, use ifStackTable. The table was mandatory for systems dividing a DS1 into channels containing different data streams that are of local interest. Systems which are indifferent to data content, such as CSUs, need not implement it. The DS1 fractional table identifies which DS1 channels associated with a CSU are being used to support a logical interface, that is, an entry in the interfaces table from the Internet-standard MIB. Consider an application managing a North American ISDN Primary Rate link whose division is a 384 kbit/s H1 _B_Channel for Video, a second H1 for data to a primary routing peer, and 1264 kbit/s H0 _B_ Channels. Consider that some subset of the H0 channels are used for voice and the remainder are available for dynamic data calls. There is a total of 14 interfaces multiplexed onto the DS1 interface. Six DS1 channels (for example, channels 1 to 6) are used for Video, six more (7 to 11 and 13) are used for data. The remaining 12 are in channels 12 and 14 to 24. If ifIndex 2 is of type DS1 and refers to the DS1 interface, and that the interfaces layered onto it are numbered 3 to 16.
	dsx3FracIfIndex.2.1 = 3
	dsx3FracIfIndex.2.2 = 3
	dsx3FracIfIndex.2.3 = 3
	dsx3FracIfIndex.2.4 = 3
	dsx3FracIfIndex.2.5 = 3
	dsx3FracIfIndex.2.6 = 3
	dsx3FracIfIndex.2.7 = 4
	dsx3FracIfIndex.2.8 = 4
	dsx3FracIfIndex.2.9 = 4
	dsx3FracIfIndex.2.10 = 4
	dsx3FracIfIndex.2.11 = 4
	dsx3FracIfIndex.2.12 = 5
	dsx3FracIfIndex.2.13 = 4
	dsx3FracIfIndex.2.14 = 6
	dsx3FracIfIndex.2.15 = 7
	dsx3FracIfIndex.2.16 = 8
	dsx3FracIfIndex.2.17 = 9
	dsx3FracIfIndex.2.18 = 10
	dsx3FracIfIndex.2.19 = 11
	dsx3FracIfIndex.2.20 = 12
	dsx3FracIfIndex.2.21 = 13
	dsx3FracIfIndex.2.22 = 14

Table 1-88 DS1-MIB Tables and Descriptions

Name	Description

	dsx3FracIfIndex.2.23 = 15
	dsx3FracIfIndex.2.24 = 16
	For North American (DS1) interfaces, there are 24 legal channels, numbered 1 through 24. For G.704 interfaces, there are 31 legal channels, numbered 1 through 31. The channels (1 to 31) correspond directly to the equivalently numbered time-slots.

DS3-MIB

The DS3-MIB describes DS3 and E3 interfaces objects.

This table lists the tables associated with this MIB.

Table 1-89 DS3-MIB Tables and Descriptions

Name	Description
dsx3ConfigTable	DS3/E3 Configuration table.
dsx3CurrentTable	DS3/E3 current table contains various statistics being collected for the current 15-minute interval.
dsx3IntervalTable	DS3/E3 Interval Table contains various statistics collected by each DS3/E3 Interface over the previous 24 hours of operation. The past 24 hours are broken into 96 completed 15-minute intervals. Each row in this table represents one such interval (identified by dsx3IntervalNumber) and for one specific interface (identified by dsx3IntervalIndex).
dsx3TotalTable	DS3/E3 Total Table contains the cumulative sum of the various statistics for the 24-hour period preceding the current interval.
dsx3FarEndConfigTable	DS3 Far End Configuration Table contains configuration information reported in the C-bits from the remote end.
dsx3FarEndCurrentTable	DS3 Far End Current table contains various statistics being collected for the current 15-minute interval. The statistics are collected from the far end block error code within the C-bits.
dsx3FarEndIntervalTable	DS3 Far End Interval Table contains various statistics collected by each DS3 interface over the previous 24 hours of operation. The past 24 hours are broken into 96 completed 15-minute intervals.
dsx3FarEndTotalTable	DS3 Far End Total Table contains the cumulative sum of the various statistics for the 24-hour period preceding the current interval.

Table 1-89 DS3-MIB Tables and Descriptions

Name	Description

dsx3FracTable	This table is deprecated in favour of using ifStackTable. Implementation of this table was optional. It was designed for those systems dividing a DS3/E3 into channels containing different data streams that are of local interest. The DS3/E3 fractional table identifies which DS3/E3 channels associated with a CSU are being used to support a logical interface, that is, an entry in the interfaces table from the Internet- standard MIB. For example, consider a DS3 device with 4 high speed links carrying router traffic, a feed for voice, a feed for video, and a synchronous channel for a non-routed protocol. We might describe the allocation of channels, in the dsx3FracTable, as follows:
	dsx3FracIfIndex.2.1 = 3
	dsx3FracIfIndex.2.2 = 3
	dsx3FracIfIndex.2.3 = 3
	dsx3FracIfIndex.2.4 = 3
	dsx3FracIfIndex.2.5 = 3
	dsx3FracIfIndex.2.6 = 3
	dsx3FracIfIndex.2.7 = 4
	dsx3FracIfIndex.2.8 = 4
	dsx3FracIfIndex.2.9 = 4
	dsx3FracIfIndex.2.10 = 4
	dsx3FracIfIndex.2.11 = 4
	dsx3FracIfIndex.2.12 = 5
	dsx3FracIfIndex.2.13 = 5
	dsx3FracIfIndex.2.14 = 5
	dsx3FracIfIndex.2.15 = 4
	dsx3FracIfIndex.2.16 = 6
	dsx3FracIfIndex.2.17 = 6
	dsx3FracIfIndex.2.18 = 6
	dsx3FracIfIndex.2.19 = 6
	dsx3FracIfIndex.2.20 = 6
	dsx3FracIfIndex.2.21 = 6
	dsx3FracIfIndex.2.22 = 6
	dsx3FracIfIndex.2.23 = 6
	dsx3FracIfIndex.2.24 = 6
	dsx3FracIfIndex.2.25 = 6
	dsx3FracIfIndex.2.26 = 6
	dsx3FracIfIndex.2.27 = 6
	dsx3FracIfIndex.2.28 = 6
	For dsx3M23, dsx3 SYNTRAN, dsx3CbitParity, and dsx3ClearChannel there are 28 legal channels, numbered 1 through 28. For e3Framed there are 16 legal channels, numbered 1 through

Table 1-89 DS3-MIB Tables and Descriptions

Name	Description
	16. The channels (1 to 16) correspond directly to the equivalently numbered time-slots.

ENTITY-MIB (RFC 2737)

The ENTITY-MIB (RFC 2737) allows functional component discovery. It is used to represent physical and logical entities (components) in the router and manages those entities. It defines managed objects for representing multiple logical entities supported by a single SNMP agent.

The entity modeling is:

- Line card port with line card as the parent
- The Xcvr container with Line card port as the parent
- If Xcvr is present, Xcvr module with Xcvr container as parent

The current software release supports the RFC 2737 version of this MIB.

The following are the conformance groups contained in the ENTITY-MIB:

- entityPhysical group–Describes the physical entities managed by a single agent.
- entityLogical group–Describes the logical entities managed by a single agent.
- entityMapping group–Describes the associations between the physical entities, logical entities, interfaces, and non-interface ports managed by a single agent.
- entityGeneral group–Describes general system attributes shared by potentially all types of entities managed by a single agent.
- entityNotifications group–Contains status indication notifications.

  The following groups are added from RFC 2737:

- entityPhysical2 group–This group augments the entityPhysical group.
- entityLogical2 group–Describes the logical entities managed by a single agent, and replaces entityLogical group.

The MIB table entPhysicalTable identifies the physical entities in the router. The entPhysicalTable contains a single row for the Cisco Carrier Routing System chassis and a row for each entity in the chassis. A physical entity may contain other entities.

The ENTITY-MIB describes a physical entity using the following information in the entPhyscialTable:

- Name–uniquely identifies the physical entity from a command console (local or virtual), or perhaps a XML-based management interface. This value must comply with the UDI Product Name guidelines [EDCS231946].
- Description–corresponds to the product description provided by CCO. This value must comply with the UDI Product Description guidelines [EDCS231946].
- Vendor Type–uniquely identifies the physical entity within an administrative domain specific to the enterprise.
- Class–indicates the class that the physical entity belongs to, including: chassis, container, power supply, fan, sensor, module, port, and cpu.
- Hardware Revision–indicates the Version IDentifier (VID) part of the Unique Device Identifier (UDI) [EDCS231946] assigned to the physical entity by manufacturing.
- Model Name–indicates the Product IDentifier (PID) part of the Unique Device Identifier (UDI) [EDCS231946] assigned to the physical entity by manufacturing. This value corresponds to the part number a customer can find on CCO for ordering.
- Serial Number–indicates the Serial Number (SN) part of the Unique Device Identifier (UDI) [EDCS231946] assigned to the physical entity by manufacturing.
- Manufacturing Name–ndicates the Top-level Assembly Number (TAN) assigned to the physical entity by manufacturing.
- CLEI URN–indicates the Common Language Equipment Identifier (CLEI) assigned to the physical entity by manufacturing, expressed as a Uniform Resource Name (URN) (RFC 4152).
- Firmware Revision–indicates the version string associated with the firmware image running on the physical entity (for example, ROMMON). If the physical entity has no associated firmware, then the value should be null-string.
- Software Revision–indicates the version string associated with the software image running on the physical entity. If the physical entity runs a modular operating system, such as IOS-XR, this value should reflect the version string associated with the main (or core) image. If the physical entity has no associated software, the value should be null-string.

- Asset Identifier—a customer assigned string-value uniquely identifying the physical entity in an administrative domain specific to that customer.
- FRU Indicator—indicates whether the physical entity is a Field Replaceable Unit (FRU).

Note This information does not apply to all classes of physical entities. See Table 1-90 for more information

This table specifies the information that applies to each class.

Table 1-90 Information the ENTITY-MIB uses from the entPhysicalTable for each class of physical entity

	Chassis	Container	Fan tray	Fan	Sensor	Module	Port	CPU
Name	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Description	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Vendor Type	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Hardware Revision	Yes	No	No	No	No	Yes	Yes	Yes
Model Name	Yes	No	No	No	No	Yes	Yes	Yes
Serial Number	Yes	No	No	No	No	Yes	Yes	Yes
Manufacturing Name	Yes	No	No	No	No	Yes	No	No
CLEI URN	Yes	No	No	No	No	Yes	No	No
Firmware Revison	No	No	No	No	No	Yes	No	No
Software Revision	No	No	No	No	No	Yes	No	No
Asset Identifier	Yes	No	No	No	No	Yes	No	No

This table lists the tables associated with the ENTITY-MIB.

Table 1-91 ENTITY-MIB Tables and Descriptions

Name	Description
entPhysicalTable	This table contains one row per physical entity. There is always at least one row for an 'overall' physical entity.
entLogicalTable	This table contains one row per logical entity. For agents that implement more than one naming scope, at least one entry must exist. Agents which instantiate all MIB objects within a single naming scope are not required to implement this table.
entLPMappingTable	This table contains zero or more rows of logical entity to physical equipment associations. For each logical entity known by this agent, there are zero or more mappings to the physical resources, which are used to realize that logical entity. An agent should limit the number and nature of entries in this table such that only meaningful and non-redundant information is returned. For example, in a system that contains a single power supply, mappings between logical entities and the power supply are not useful and should not be included. Also, only the most appropriate physical component, which is closest to the root of a particular containment tree, should be identified in an entLPMapping entry. For example, suppose a bridge is realized on a particular module, and all ports on that module are ports on this bridge. A mapping between the bridge and the module would be useful, but additional mappings between the bridge and each of the ports on that module would be redundant (because the entPhysicalContainedIn hierarchy can provide the same information). On the other hand, if more than one bridge were utilizing ports on this module, mappings between each bridge and the ports it used would be appropriate. Also, in the case of a single backplane repeater, a mapping for the backplane to the single repeater entity is not necessary.
entAliasMappingTable	This table contains zero or more rows, representing mappings of logical entity and physical component to external MIB identifiers. Each physical port in the system may be associated with a mapping to an external identifier, which itself is associated with a particular logical entity's naming scope. A 'wildcard' mechanism is provided to indicate that an identifier is associated with more than one logical entity.
entPhysicalContainsTable	Table that exposes the container/containee relationships between physical entities. This table provides all the information found by constructing the virtual containment tree for a given entPhysicalTable, but in a more direct format. In the event a physical entity is contained by more than one other physical entity (for example, double-wide modules), this table should include these additional mappings, which cannot be represented in the entPhysicalTable virtual containment tree.

MIB Constraints

This table lists the constraints on objects in the ENTITY-MIB.

Table 1-92 ENTITY-MIB Constraints

MIB Object	Notes
entPhysicalTable	SNMP sets are not supported. Unable to show information for powered down LC modules. No entry for preconfigured interfaces.
cefcFRUPowerStatusTable	SNMP sets not supported. (cefcFRUPowerAdminStatus)
entModuleTable	SNMP sets not supported. (cefcModuleAdminStatus)
entLogicalTable	entLogicalType not supported.
entLPMpapingTable	
entLogicalCommunity	Not supported.
entLogicalTAddress	Not supported.
entLogicalTDomain	Not supported.
entLogicalContextEngineID	Not supported.
entLogicalContextName	Not supported.

ENTITY-STATE-MIB

The ENTITY-STATE-MIB defines a state extension to the Entity MIB. Copyright (C) The Internet Society 2005. This version of this MIB module is part of RFC 4268; see the RFC itself for full legal notices.

This table lists the tables associated with this MIB.

Table 1-93 ENTITY-STATE-MIB Tables and Descriptions

Name	Description
entStateTable	A table of information about state/status of entities. This is a sparse augment of the entPhysicalTable. Entries appear in this table for values of entPhysicalClass [RFC4133] that in this implementation are able to report any of the state or status stored in this table

MIB Constraints

This table lists the constraints on objects in the ENTITY-STATE-MIB.

Table 1-94 ENTITY-STATE-MIB Constraints

MIB Object	Notes
entStateLastChanged	Supported
entStateAdmin	Read-only
entStateOper	Supported
entStateUsage	Not Supported
entStateAlarm	Not Supported
entStateStandby	Supported

EVENT-MIB

The EVENT-MIB contains objects to define event triggers and actions for network management purposes.

This table lists the tables associated with this MIB.

Table 1-95 EVENT-MIB Tables and Descriptions

Name	**Description**
mteTriggerTable	Table of management event trigger information
mteTriggerDeltaTable	Table of management event trigger information for delta sampling
mteTriggerExistenceTable	Table of management event trigger information for existence triggers
mteTriggerBooleanTable	Table of management event trigger information for boolean triggers
mteTriggerThresholdTable	Table of management event trigger information for threshold triggers
mteObjectsTable	Table of objects that can be added to notifications based on the trigger, trigger test, or event, as pointed to by entries in those tables
mteEventTable	Table of management event action information
mteEventNotificationTable	Table of information about notifications to be sent as a consequence of management events
mteEventSetTable	Table of management event action information

MIB Constraints

This table lists the constraints on objects in the EVENT-MIB.

Table 1-96 EVENT-MIB Constraints

MIB Object	**Notes**
mteTriggerDeltaDiscontinuityID	Not supported.
mteTriggerDeltaDiscontinuityIDWildcard	Not supported.
mteTriggerDeltaDiscontinuityIDType	Not supported.

EXPRESSION-MIB

The EXPRESSION-MIB defines expressions of MIB objects for network management purposes. This MIB is an early snapshot of work done by the IETF Distributed Management working group. After this snapshot was taken, the MIB was modified, had new OIDs assigned, and then published as RFC 2982.

This table lists the tables associated with this MIB.

Table 1-97 EXPRESSION-MIB Tables and Descriptions

Name	**Description**
expNameTable	Table of expression names, for creating and deleting expressions

Table 1-97 EXPRESSION-MIB Tables and Descriptions

Name	Description
expExpressionTable	Table of expression definitions
expObjectTable	Table of object definitions for each expExpression. Wildcarding instance IDs: It is legal to omit all or part of the instance portion for some or all of the objects in an expression. (See the description of expObjectID for details). However, note that if more than one object in the same expression is wildcarded in this way, they all must be objects where that portion of the instance is the same. In other words, all objects may be in the same sequence or in different sequences but with the same semantic index value (that is, a value of ifIndex) for the wildcarded portion.
expValueTable	Table of values from evaluated expressions

FRAME-RELAY-DTE-MIB

The FRAME-RELAY-DTE-MIB describes the use of a Frame Relay interface by a DTE.

This table lists the tables associated with this MIB.

Table 1-98 FRAME-RELAY-DTE-MIB Tables and Descriptions

Name	Description
frDlcmiTable	Parameters for the Data Link Connection Management Interface for the frame relay service on this interface.
frCircuitTable	Table containing information about specific DLC[1] or virtual circuits.
frErrTable	Table containing information about Errors on the Frame Relay interface. Discontinuities in the counters contained in this table are the same as apply to the ifEntry associated with the Interface.

[1] DLC = data link connections

IEEE8021-CFM-MIB

The IEEE8021-CFM-MIB is a Connectivity Fault Management (CFM) module for managing IEEE 802.1ag.

MIB Objects

This table lists the tables associated with this MIB.

Table 1-99 IEEE8021-CFM-MIB Tables and Descriptions

Name	Description
OBJECT-TYPE	There is one CFM Stack table per bridge. It permits the retrieval of information about the Maintenance Points configured on any given interface.
OBJECT-TYPE	This table defines the association of VIDs into VLANs. There is an entry in this table, for each component of the bridge, for each VID that is: a) a VID belonging to a VLAN associated with more than one VID; and b) not the Primary VLAN of that VID.
OBJECT-TYPE	For each bridge component, the Default MD Level Managed Object

Table 1-99 *IEEE8021-CFM-MIB Tables and Descriptions*

Name	Description
	controls MHF creation for VIDs that are not attached to a specific Maintenance Association Managed Object, and Sender ID TLV transmission by those MHFs.
OBJECT-TYPE	The CFM Configuration Error List table provides a list of Interfaces and VIDs that are incorrectly configured.
OBJECT-TYPE	The Maintenance Domain table. Each row in the table represents a different Maintenance Domain. A Maintenance Domain is described in 802.1ag (3.21) as the network or the part of the network for which faults in connectivity are to be managed.
OBJECT-TYPE	The Maintenance Association table. Each row in the table represents an MA. An MA is a set of MEPs, each configured with a single service instance.
OBJECT-TYPE	The Maintenance Association table. Each row in the table represents an MA. An MA is a set of MEPs, each configured with a single service instance. This is the part of the complete MA table that is variable across the Bridges in a Maintenance Domain, or across the components of a single Bridge.
OBJECT-TYPE	List of MEPIDs that belong to this MA. Clause 12.14.6.1.3 specifies that a list of MEPIDs in all bridges in that MA, but since SNMP SMI does not allow to state in a MIB that an object in a table is an array, the information has to be stored in another table with two indices, being the first index, the index of the table that contains the list or array.
OBJECT-TYPE	The Maintenance Association End Point (MEP) table. Each row in the table represents a different MEP. A MEP is an actively managed CFM entity, associated with a specific DSAP of a Service Instance, which can generate and receive CFM PDUs and track any responses.
OBJECT-TYPE	This table extends the MEP table and contains a list of Linktrace replies received by a specific MEP in response to a linktrace message.
OBJECT-TYPE	The MEP Database. A database, maintained by every MEP, that maintains received information about other MEPs in the Maintenance Domain.

MIB Constraints

This table lists the constraints on objects in the IEEE8021-CFM-MIB.

Table 1-100 *IEEE8021-CFM-MIB Constraints*

MIB Object	Notes
dot1agCfmVlanPrimaryVid	Not supported
dot1agCfmVlanRowStatus	Not supported
dot1agCfmMdName	Not supported
dot1agCfmMdFormat	Not supported
dot1agCfmMdMdLevel	Not supported
dot1agCfmMdMhfCreation	Not supported
dot1agCfmMdMhfIdPermission	Not supported

Table 1-100 ***IEEE8021-CFM-MIB Constraints***

MIB Object	Notes
dot1agCfmMdRowStatus	Not supported
dot1agCfmDefaultMdDefLevel	Not supported
dot1agCfmDefaultMdDefMhfCreation	Not supported
dot1agCfmDefaultMdDefIdPermission	Not supported
dot1agCfmDefaultMdLevel	Not supported
dot1agCfmDefaultMdMhfCreation	Not supported
dot1agCfmDefaultMdIdPermission	Not supported
dot1agCfmMepIfIndex	Not supported
dot1agCfmMepDirection	Not supported
dot1agCfmMepPrimaryVid	Not supported
dot1agCfmMepActive	Not supported
dot1agCfmMepCciEnabled	Not supported
dot1agCfmMepCcmLtmPriority	Not supported
dot1agCfmMepLowPrDef	Not supported
dot1agCfmMepFngAlarmTime	Not supported
dot1agCfmMepFngResetTime	Not supported
dot1agCfmMepTransmitLbmStatus	Not supported
dot1agCfmMepTransmitLbmDestMacAddress	Not supported
dot1agCfmMepTransmitLbmDestMepId	Not supported
dot1agCfmMepTransmitLbmDestIsMepId	Not supported
dot1agCfmMepTransmitLbmMessages	Not supported
dot1agCfmMepTransmitLbmDataTlv	Not supported
dot1agCfmMepTransmitLbmVlanPriori ty	Not supported
dot1agCfmMepTransmitLbmVlanDropEnable	Not supported
dot1agCfmMepTransmitLtmFlags	Not supported
dot1agCfmMepTransmitLtmTargetMacAddress -	Not supported
dot1agCfmMepTransmitLtmTargetMep- Id	Not supported
dot1agCfmMepTransmitLtmTargetIsMe- pId	Not supported
dot1agCfmMepTransmitLtmTtl	Not supported
dot1agCfmMepTransmitLtmEgressId	Not supported

Table 1-100 IEEE8021-CFM-MIB Constraints

MIB Object	Notes
en- tifier	
dot1agCfmMepRowStatus	Not supported

IEEE8023-LAG-MIB

The IEEE8023-LAG-MIB provides access to the administrative, operational and diagnostics state for bundles and ports operating the IEEE 802.3ad Link Aggregation Control Protocol.

This table lists the tables associated with this MIB.

Table 1-101 IEEE8023-LAG-MIB Tables and Descriptions

Name	**Description**
dot3adAggTable	Table that contains information about every Aggregator running the IEEE 802.3ad Link Aggregation Control Protocol that is associated with this System
dot3adAggPortListTable	Table that contains a list of all the ports associated with each Aggregator running the IEEE 802.3ad Link Aggregation Control Protocol.
dot3adAggPortTable	Table that contains Link Aggregation Control configuration information about every Aggregation Port running the IEEE 802.3ad Link Aggregation Control Protocol associated with this device. A row appears in this table for each physical port
dot3adAggPortStatsTable	Table that contains Link Aggregation information about every port running the IEEE 802.3ad Link Aggregation Control Protocol that is associated with this device. A row appears in this table for each physical port
dot3adAggPortDebugTable	Table that contains Link Aggregation debug information about every port running the IEEE 802.3ad Link Aggregation Control Protocol that is associated with this device. A row appears in this table for each physical port
dot3adTablesLastChanged	This object indicates the time of the most recent change to the dot3adAggTable, dot3adAggPortListTable or dot3AggPortTable.

MIB Constraints

This table lists the constraints on objects in the IEEE8023-LAG-MIB.

Table 1-102 IEEE8023-LAG-MIB Constraints

MIB Object	**Notes**
dot3adAggPortListTable	dot3adAggPortListPorts is not supported.

IF-MIB (RFC 2863)

The IF-MIB (RFC 2863) describes the attributes of physical and logical interfaces (network interface sublayers). The router supports the ifGeneralGroup of MIB objects for all layers (ifIndex, ifDescr, ifType, ifSpeed, ifPhysAddress, ifAdminStatus, ifOperStatus, ifLastChange, ifName, ifLinkUpDownTrapEnable, and ifHighSpeed).

One of the most commonly used identifiers in SNMP-based network management applications is the Interface Index (ifIndex) value. IfIndex is a unique identifying number associated with a physical or logical interface. For more information about SNMP commands, see http://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k_r4-

2/system_management/command_reference/b_sysman_cr42asr9k/b_sysman_cr42asr9k_chapter_01101.html

To provide access to information on selected interfaces stored in IF-MIB table to a user, this MIB is made vrf-aware. This makes management of IF-MIB table for VRF based networks more secure. Context based community can be used only when VRF based polling needs to be done.

This table lists the tables associated with this MIB.

Note The object ifNumber is not commited on SNMP Data Collection Manager (DCM).

Table 1-103 IF-MIB Tables and Descriptions

Name	Description
ifTable	List of interface entries. The number of entries is given by the value of ifNumber. This table is vrf-aware.
ifXtable	List of interface entries. The number of entries is given by the value of ifNumber. This table contains additional objects for the interface table. This table is vrf-aware.
ifStackTable	Table containing information on the relationships between the multiple sub-layers of network interfaces. In particular, it contains information on which sub-layers run 'on top of' which other sub-layers, where each sub-layer corresponds to a conceptual row in the ifTable. For example, when the sub-layer with ifIndex value x runs over the sub-layer with ifIndex value y, then this table contains: ifStackStatus.x.y=active For each ifIndex value, I, which identifies an active interface, there are always at least two instantiated rows in this table associated with I. For one of these rows, I is the value of ifStackHigherLayer; for the other, I is the value of ifStackLowerLayer. (If I is not involved in multiplexing, these are the only two rows associated with I.) For example, two rows exist even for an interface which has no others stacked on top or below it: ifStackStatus.0.x=active ifStackStatus.x.0=active
ifRcvAddressTable	This table contains an entry for each address (broadcast, multicast, or unicast) for which the system receives packets/frames on a particular interface, except as follows: ■ for an interface operating in promiscuous mode, entries are only required for those addresses for which the system would receive frames were it not operating in promiscuous mode. ■ for 802.5 functional addresses, only one entry is required, for the address which has the functional address bit ANDed with the bit mask of all functional addresses for which the interface accepts frames. A system is normally able to use any unicast address, which corresponds to an entry in this table as a source address.
ifTestTable	This table contains one entry per interface. It defines objects which allow a network manager to instruct an agent to test an interface for various faults. Tests for an interface are defined in the media-specific MIB for that interface. After invoking a test, the object ifTestResult can be read to determine the outcome. If an agent can not perform the test, ifTestResult is set to so indicate. The object ifTestCode can be used to provide further test-specific or interface-specific (or even enterprise-specific) information concerning the outcome of the test. Only one test can be in progress on each interface at any one time. If one test is in progress when another test is invoked, the second test is rejected.

Table 1-103 IF-MIB Tables and Descriptions

Name	Description
	Some agents may reject a test when a prior test is active on another interface.

IP-FORWARD-MIB

The IP-FORWARD-MIB describes the control of the display of Classless Interdomain Routing (CIDR) multipath IP routes (IPv4 - RFC 2096) and the management of CIDR IP routes (IPv6 - RFC 4292).

To provide selective access to information stored in IP forwarding table to user, the IP-FORWARD-MIB is made vrf-aware. This makes management of IP forwarding table for VRF based networks more secure.

This table lists the tables associated with this MIB.

Table 1-104 IP-FORWARD-MIB Tables and Descriptions

Name	Description
inetCidrRouteTable	This entity IP Routing table (when MIB is used to poll IPv6 route information).
ipCidrRouteTable	This entity IP Routing table. This table has been deprecated in favor of the IP version neutral inetCidrRouteTable.
ipForwardTable	This entity’s IP Routing table. The ipForwardNumber object is vrf-aware.

IP-MIB

The IP-MIB contains objects for managing IP and Internet Control Message Protocol (ICMP) implementations.

Note The IP-MIB does not provide functionality to manage IP routes.

This table lists the tables associated with this MIB.

Table 1-105 IP-MIB Tables and Descriptions

Name	Description
ipv4InterfaceTable	Not supported
ipv6InterfaceTable	Table containing per-interface IPv6-specific information
ipSystemStatsTable	Table containing system wide, IP version specific traffic statistics. This table and the ipIfStatsTable contain similar objects whose difference is in their granularity. Where this table contains system wide traffic statistics, the ipIfStatsTable contains the same statistics but counted on a per-interface basis.
ipIfStatsTable	Table containing per-interface traffic statistics. This table and the ipSystemStatsTable contain similar objects whose difference is in their granularity. Where this table contains per-interface statistics, the ipSystemStatsTable contains the same statistics, but counted on a system wide basis. This table is only applicable to IPv6, there is no support available for IPv4 stats.
ipAddressPrefixTable	Table allows the user to determine the source of an IP address or set of IP addresses, and allows other tables to share the information via pointer rather than by copying. For example, when the node configures a unicast and anycast address for a prefix, the ipAddressPrefix objects

Table 1-105 IP-MIB Tables and Descriptions

Name	Description
	for those addresses point to a single row in this table. This table primarily provides support for IPv6 prefixes, and several of the objects are less meaningful for IPv4. The table continues to allow IPv4 addresses to allow future flexibility. To promote a common configuration, this document includes suggestions for default values for IPv4 prefixes. Each of these values may be overridden if an object is meaningful to the node. All prefixes used by this entity should be included in this table independent of how the entity learned the prefix. (This table is not limited to prefixes learned from router advertisements.
ipAddressTable	This table contains addressing information relevant to the entity's interfaces. This table does not contain multicast address information. Tables for such information should be contained in multicast specific MIBs, such as RFC 3019. While this table is writable, note that several objects, such as ipAddressOrigin, are not. The intention in allowing a user to write to this table is to allow them to add or remove any entry that is not permanent. The user should be allowed to modify objects and entries when that would not cause inconsistencies within the table. Allowing write access to objects, such as ipAddressOrigin, could allow a user to insert an entry and then label it incorrectly. **Note** When including IPv6 link-local addresses in this table, the entry must use an InetAddressType of 'IPv6z' in order to differentiate between the possible interfaces.
ipNetToPhysicalTable	IP Address Translation table used for mapping from IP addresses to physical addresses. The Address Translation tables contain the IP address to 'physical' address equivalences. Some interfaces do not use translation tables for determining address equivalences (that is, DDN-X.25 has an algorithmic method); if all interfaces are of this type, the Address Translation table is empty, that is, has zero entries. While many protocols may be used to populate this table, ARP and Neighbor Discovery are the most likely options
ipv6ScopeZoneIndexTable	Not supported
ipDefaultRouterTable	Not supported
ipv6RouterAdvertTable	Not supported
icmpStatsTable	Table of generic system-wide ICMP counters
icmpMsgStatsTable	Table of system-wide per-version, per-message type ICMP counters
ipAddrTable	Table of addressing information relevant to this entity's IPv4 addresses. This table has been deprecated, as a new IP version-neutral table has been added. It is loosel replaced by the ipAddressTable although several objects that were not deemed useful were not carried forward while another (ipAdEntReasmMaxSize) was moved to the ipv4InterfaceTable
ipNetToMediaTable	IPv4 Address Translation table used for mapping from IPv4 addresses to physical addresses. This table has been deprecated, as a new IP version-neutral table has been added. It is loosely replaced by the

Table 1-105 IP-MIB Tables and Descriptions

Name	Description
	ipNetToPhysicalTable

MIB Constraints

This table lists the constraints that the router places on objects in the IP-MIB. For detailed definitions of MIB objects, see the MIB.

Note Tables which are specific to IPv4 are not implemented.

Table 1-106 IP-MIB Constraints

MIB Object	Notes
ipv6InterfaceTable	
■ ipv6InterfaceIdentifier	Lower n bits of link local address, where n=128 - prefix-len
■ ipv6InterfaceEnableStatus	up (1) if link local address is configured else down (2)
■ ipv6InterfaceForwarding	ipv6InterfaceForwarding(1) if IPv6 is configured on LC or notForwarding(2) if IPv6 is configured on RP
ipSystemStatsTable	
■ ipSystemStatsinOctets	Not supported
■ ipSystemStatsinNoRoutes	Not supported
■ ipSystemStatsInAddrErrors	Not supported
■ ipSystemStatsInDiscards	Not supported
■ ipSystemStatsOutNoRoutes	Not supported
■ ipSystemStatsOutForwDatagrams	Not supported
■ ipSystemStatsOutDiscards	Not supported
■ ipSystemStatsOutTransmits	Not supported
■ ipSystemStatsOutOctets	Not supported
■ ipSystemStatsInMcastPkts	Not supported
■ ipSystemStatsInMcastOctets	Not supported
■ ipSystemStatsOutMcastPkts	Not supported
■ ipSystemStatsOutMcastOctets	Not supported
■ ipSystemStatsInBcastPkts	Not supported
■ ipSystemStatsOutBcastPkts	Not supported
■ ipSystemStatsDiscontinuityTime	Not supported

Table 1-106 IP-MIB Constraints

MIB Object	Notes
ipIfStatsTable	Table only applicable to IPv6
■ ipIfStatsRefreshRate	Not supported
■ Protocol related counters on per-interface basis (22 objects in this table)	Not supported
ipAddressPrefixTable	
■ ipAddressPrefixPrefix	First n bits of ipv6_addr () where n=prefix_len
■ ipAddressPrefixLength	prefix_len field
■ ipAddressPrefixOrigin	manual(2) if prefix is taken from global address or wellknown(3) if prefix is taken from link local address
■ ipAddressPrefixAutonomousFlag	False(2) for link local and True(1) for others
■ ipAddressPrefixAdvPreferredLifetime	Not supported
■ ipAddressPrefixAdvValidLifetime	Not supported
ipAddressTable	
■ ipAddressPrefix	First n bits of ipv6_addr () where n=prefix_len
■ ipAddressOrigin	‘manual’ is address if global or ‘linklayer’ if it is link local
■ ipAddressCreated	Not supported
■ ipAddressLastChanged	Not supported
ipv6ScopeZoneIndexTable	Not supported
ipDefaultRouterTable	Not supported
ipRouterAdvertTable	Not supported
icmpStatsTable	
■ icmpStatsOutErrors	Not supported
Scalar Objects	
■ ipv6InterfaceTableLastChange	Not supported
■ ipv6IpDefaultHopLimit	Not supported

IPV6-MIB

The IPV6-MIB describes the entities implementing the IPV6 protocol.

This table lists the tables associated with this MIB.

Table 1-107 IPV6-MIB Tables and Descriptions

Name	Description
ipv6IfTable	IPv6 Interfaces table contains information on the entity's internetwork-layer interfaces. An IPv6 interface constitutes a logical network layer attachment to the layer immediately below IPv6 including internet layer 'tunnels', such as tunnels over IPv4 or IPv6 itself
ipv6IfStatsTable	IPv6 interface traffic statistics
ipv4InterfaceTable	Not supported
ipDefaultRouterTable	Not supported
ipv6ScopeZoneIndexTable	Not supported
ipv6AddrPrefixTable	List of IPv6 address prefixes of IPv6 interfaces
ipv6AddrTable	Table of addressing information relevant to this nodes interface addresses
ipv6RouteTable	Not supported
ipv6NetToMediaTable	IPv6 Address Translation table used for mapping from IPv6 addresses to physical addresses. The IPv6 address translation table contain the Ipv6Address to physical address equivalencies. Some interfaces do not use translation tables for determining address equivalencies; if all interfaces are of this type, then the Address Translation table is empty, that is, has zero entries

MIB Constraints

This table lists the constraints that the router places on objects in the IPV6-MIB. For detailed definitions of MIB objects, see the MIB.

Table 1-108 IPV6-MIB Constraints

Name	Description
ipDefaultRouterTable	Not supported
ipv6ScopeZoneIndexTable	Not supported
ipv4InterfaceTable	Not supported
ipIfStatsTableLastChange	Not supported

IPV6-MLD-MIB

The IPV6-MLD-MIB is the MIB module for MLD management.

This table lists the tables associated with this MIB.

Table 1-109 ***IPV6-MLD-MIB Tables and Descriptions***

Name	Description
mldInterfaceTable	(conceptual) Table listing the interfaces on which MLD is enabled
mldCacheTable	(conceptual) Table listing the IPv6 multicast groups for which there are members on a particular interface

IPV6-TC

The IPV6-TC contains TCs for IPV6. There are no tables associated with this MIB.

ISIS-MIB

The IS-IS MIB describes a management information base for the IS-IS Routing protocol, as described in ISO 10589, when it is used to construct routing tables for IP networks, as described in RFC 1195.

This table lists the tables associated with this MIB.

Table 1-110 ***ISIS-MIB Tables and Descriptions***

Name	Description
isisManAreaAddrTable	Set of manual area addresses configured on this Intermediate System. At least one row in which the value of isisManAreaAddrExistState is active must be present. The maximum number of rows in this table for which the object isisManAreaAddrExistState has the value active is three. An attempt to create more than three rows of isisManAreaAddrEntry with state 'active' in one instance of the IS-IS protocol should return inconsistentValue
isisAreaAddrTable	Union of the sets of area addresses reported in all Level 1 LSPs with fragment number zero generated by this Intermediate System, or received from other Intermediate Systems that are reachable via Level 1 routing
isisRedistributeAddrTable	This table provides criteria to decide if a route should be leaked from Layer 2 to Layer 1 when Domain Wide Prefix leaking is enabled. Addresses that match the summary mask in the table MUST be announced at Layer 1 by routers when isisSysL2toL1Leaking is enabled. Routes that fall into the ranges specified are announced as is, without being summarized. Routes that do not match a summary mask are not announced
isisRouterTable	Set of hostnames and router ID
isisSysLevelTable	Level specific information about the System
isisCircTable	The table of circuits used by this Intermediate System
isisCircLevelTable	Level specific information about circuits used by IS-IS
isisSystemCounterTable	System-wide counters for this Intermediate System

Table 1-110 ***ISIS-MIB Tables and Descriptions***

Name	**Description**
isisCircuitCounterTable	Circuit specific counters for this Intermediate System
isisPacketCounterTable	Information about IS-IS protocol traffic at one level, on one circuit, in one direction
isisISAdjTable	Table of adjacencies to Intermediate Systems
isisISAdjAreaAddrTable	This table contains the set of Area Addresses of neighboring Intermediate Systems as reported in received IIH PDUs
isisISAdjIPAddrTable	This table contains the set of IP Addresses of neighboring Intermediate Systems as reported in received IIH PDUs
isisISAdjProtSuppTable	This table contains the set of protocols supported by neighboring Intermediate Systems as reported in received IIH PDUs
isisRATable	Table of Reachable Addresses to NSAPs or Address Prefixes
isisIPRATable	Table of IP Reachable Addresses to networks, subnetworks, or hosts either manually configured or learned from another protocol
isisLSPSummaryTable	Table of LSP Headers
isisLSPTLVTable	Table of LSPs in the database

MIB Constraints

This table lists the constraints that the router places on objects in the ISIS-MIB. For detailed definitions of MIB objects, see the MIB.

Note SNMP sets are not supported.

Table 1-111 ***ISIS-MIB Constraints***

MIB Object	**Notes**
isisAreaAddrTable	isisAreaAddr not supported
isisCircuitGroup	Not supported
isisISAdjGroup	Not supported
isisISIPRADestGroup	Not supported
isisLSPGroup	Not supported
isisManAreaAddrTable	isisManAreaAddrExistState not supported
isisNotificationTable	Not supported
isisRATable Group	Not supported
isisRedistributeAddrEntry	isisRedistributeAddrExistState
isisRouterTable	isisRouterID is not supported
isisSummAddrTable	
isisSummAddrExistState	Not supported
isisSummAddrMetric	Not supported
isisSummAddrFullMetric	Not supported

MEF-SOAM-PM-MIB

This MIB module contains the management objects for the management of Ethernet Services Operations, Administration and Maintenance for Performance Monitoring.

This table lists the tables associated with this MIB.

Name	Description
mefSoamPmMepTable	This table is an extension of the dot1agCfmMepTable and rows are automatically added or deleted from this table based upon row creation and destruction of the dot1agCfmMepTable. This table represents the local MEP PM configuration table. The primary purpose of this table is provide local parameters for the SOAM PM function found in [Y.1731] and [MEF SOAM-PM] and instantiated at a MEP.
mefSoamLmCfgTable	This table includes configuration objects and operations for the Frame Loss Measurement function defined in [Y.1731] and [MEF SOAM-PM]. Each row in the table represents a Loss Measurement session for the defined MEP. This table uses four indices. The first three indices are the indices of the Maintenance Domain, MaNet, and MEP tables. The fourth index is the specific LM session on the selected MEP. A Loss Measurement session is created on an existing MEP by first accessing the mefSoamPmMepOperNextIndex object and using this value as the mefSoamLmCfgIndex in the row creation. Some writable objects in this table are only applicable in certain cases (as described under each object), and attempts to write values for them in other cases will be ignored. The writable objects in this table need to be persistent upon reboot or restart of a device.
mefSoamLmMeasuredStatsTable	This object contains the last measured results for a SOAM Loss Measurement session. Each row in the table represents a Loss Measurement session for the defined MEP. This table uses four indices. The first three indices are the indices of the Maintenance Domain, MaNet, and MEP tables. The fourth index is the specific LM session on the selected MEP. Instances of this managed object are created automatically by the SNMP Agent when the Loss Measurement session is running. Each object in this table applies only if the corresponding bit is set in mefSoamLmCfgMeasurementEnable. The objects in this table do not need to be persistent upon reboot or restart of a device.
mefSoamLmCurrentAvailStatsTable	This object contains the current results for a SOAM Loss Measurement session for availability statistics gathered during the interval indicated by mefSoamLmCfgAvailabilityMeasurementInterval. Each row in the table represents a Loss Measurement session for the defined MEP. This table uses four indices. The first three indices are the indices of the Maintenance Domain, MaNet, and MEP tables. The fourth index is the specific LM session on the selected MEP. Instances of this managed object are created automatically by the SNMP Agent when the Loss Measurement session is running. The objects in this table apply regardless of the value of mefSoamLmCfgType unless otherwise specified in the object description. Except for mefSoamLmCurrentAvailStatsIndex, mefSoamLmCurrentAvailStatsStartTime, mefSoamLmCurrentAvailStatsElapsedTime and mefSoamLmCurrentAvailStatsSuspect, each object in this table applies only if the corresponding bit is set in mefSoamLmCfgMeasurementEnable. The objects in this table may be persistent upon reboot or restart of a device.

Name	Description
mefSoamLmCurrentStatsTable	This table contains the results for the current Measurement Interval in a SOAM Loss Measurement session gathered during the interval indicated by mefSoamLmCfgMeasurementInterval. A row in this table is created automatically by the SNMP Agent when the Loss Measurement session is configured. Each row in the table represents the current statistics for a Loss Measurement session for the defined MEP. This table uses four indices. The first three indices are the indices of the Maintenance Domain, MaNet, and MEP tables. The fourth index is the specific LM session on the selected MEP. There may be more than one LM session per MEP. The main use case for this is to allow multiple CoS instances to be operating simultaneously for a MEP. The objects in this table apply regardless of the value of mefSoamLmCfgType unless otherwise specified in the object description. Except for mefSoamLmCurrentStatsIndex, mefSoamLmCurrentStatsStartTime, mefSoamLmCurrentStatsElapsedTime and mefSoamLmCurrentStatsSuspect, each object in this table applies only if the corresponding bit is set in mefSoamLmCfgMeasurementEnable. The objects in this table do not need to be persistent upon reboot or restart of a device.
mefSoamLmHistoryAvailStatsTable	This table contains the results for availability history Measurement Intervals in a SOAM Loss Measurement session. Rows of this table object are created automatically by the SNMP Agent when the Loss Measurement session is running and a Measurement Interval is completed. Each row in the table represents the history statistics for a Loss Measurement session availability Measurement Interval for the defined MEP. This table uses five indices. The first three indices are the indices of the Maintenance Domain, MaNet, and MEP tables. The fourth index is the specific LM session on the selected MEP. The fifth index index the specific Measurement Interval. At least 32 completed Measurement Intervals are to be supported. 96 completed Measurement Intervals are recommended to be supported. If there are at least 32 rows in the table and a new Measurement Interval completes and a new row is to be added to the table, the oldest completed Measurement Interval can be deleted (row deletion). If the availability Measurement Interval is other than 15 minutes then a minimum of 8 hours of completed Measurement Intervals are to be supported and 24 hours are recommended to be supported. Except for mefSoamLmHistoryAvailStatsIndex,mefSoamLmHistoryAvailStatsEndTime, mefSoamLmHistoryAvailStatsElapsedTime and mefSoamLmHistoryAvailStatsSuspect, each object in this table applies only if the corresponding bit is set in mefSoamLmCfgMeasurementEnable. The rows and objects in this table are to be persistent upon reboot or restart of a device.
mefSoamLmHistoryStatsTable	This table contains the results for history Measurement Intervals in a SOAM Loss Measurement session. Rows of this table object are created automatically by the SNMP Agent when the Loss Measurement session is running and a Measurement Interval is completed. Each row in the table represents the history statistics for a Loss Measurement session Measurement Interval for the defined MEP. This table uses five indices. The first three indices are the indices of the Maintenance Domain, MaNet, and MEP tables. The fourth index is the specific LM session on the selected MEP. The fifth index index the specific Measurement Interval. At least 32 completed Measurement Intervals are to be supported. 96 completed Measurement Intervals are recommended to be supported. If there are at least 32 rows in the

Name	Description
	table and a new Measurement Interval completes and a new row is to be added to the table, the oldest completed Measurement Interval may be deleted (row deletion). If the measurement interval is other than 15 minutes then a minimum of 8 hours of completed Measurement Intervals are to be supported and 24 hours are recommended to be supported. Except for mefSoamLmHistoryStatsIndex, mefSoamLmHistoryStatsEndTime, mefSoamLmHistoryStatsElapsedTime and mefSoamLmHistoryStatsSuspect, each object in this table applies only if the corresponding bit is set in mefSoamLmCfgMeasurementEnable. The rows and objects in this table are to be persistent upon reboot or restart of a device.
mefSoamDmCfgTable	This table includes configuration objects and operations for the Delay Measurement function. Each row in the table represents a Delay Measurement session for the defined MEP. This table uses four indices. The first three indices are the indices of the Maintenance Domain, MaNet, and MEP tables. The fourth index is the specific DM session on the selected MEP. A Delay Measurement session is created on an existing MEP by first accessing the mefSoamDmOperNextIndex object and using this value as the mefSoamDmCfgIndex in the row creation. Some writable objects in this table are only applicable in certain cases (as described under each object), and attempts to write values for them in other cases will be ignored. The writable objects in this table need to be persistent upon reboot or restart of a device.
mefSoamDmCfgMeasBinTable	This table includes configuration objects for the Delay Measurement bins to collect stats. Each row in the table is automatically created when the Delay Measurement session is defined for the selected MEP. The number of rows created is based upon three items: the DM type, the number of bins defined for each type, and whether bins are enabled for each type. The first four indices are the same as used to create the DM session: Maintenance Domain, MaNet, MEP identification, and mefSoamDmCfgIndex. The fifth index is the type of bin, and the sixth index is the bin number. For a dmDmm session all nine types of bins can be created. For a dm1DmmTx session no bins are created. For a dm1DmmRx session only type’s forwardFrameDelay, forwardIfdv, and forwardFrameDelayRange can be created. The number of bins created for a bin type is based upon: the mefSoamDmCfgNumMeasBinsPerFrameDelayIn terval object, the mefSoamDmCfgNumMeasBinsPerInterFrameDe layVariationInterval object, the mefSoamDmCfgNumMeasBinsPerFrameDelayRangeInterval object, and the mefSoamDmCfgMeasurementEnable object. For instance, if a dmDmm session with Bins per Frame Delay Interval set to 5, Bins per Frame Delay Variation Interval set to 3, and Frame Delay Range set to 2 (default), all of the appropriate bits set in mefSoamDmMeasurementCfgEnable, the following number of rows would be created: For bin types TwoWayFrameDelay(1), forwardFrameDelay(2), and backwardFrameDelay(3) = 5 bins * 3 types = 15 rows For bin types TwoWayIfdv(4), forwardIfdv(5), backwardIfdv(6) = 3 bins * 3 types = 9 rows. For bins types

Name	Description
	twoWayFrameDelayRange(7), forwardFrameDelayRange(8), backwardFrameDelayRange(9) = 2 bins * 3 types = 6 rows. This gives a total of 30 rows created for the DMM session example. Each value in the bin defaults to 5000us greater than the previous bin, with the first bin default value set to 0. For the delay example above (5 bins), the following default values would be written to the bins: bin 1:0 (range is 0us).

MPLS-L3VPN-STD-MIB

The MPLS-L3VPN-STD-MIB contains managed object definitions for the Layer-3 Multiprotocol Label Switching Virtual Private Networks.

This table lists the tables associated with this MIB.

Table 1-112 MPLS-L3VPN-STD-MIB Tables and Descriptions

Name	Description
mplsL3VpnIfConfTable	This table specifies per-interface MPLS capability and associated information
mplsL3VpnVrfTable	This table specifies per-interface MPLS L3VPN VRF Table capability and associated information. Entries in this table define VRF routing instances associated with MPLS/VPN interfaces. Note that multiple interfaces can belong to the same VRF instance. The collection of all VRF instances comprises an actual VPN
mplsL3VpnVrfRTTable	This table specifies per-VRF route target association. Each entry identifies a connectivity policy supported as part of a VPN
mplsL3VpnVrfSecTable	This table specifies per MPLS L3VPN VRF Table security-related counters
mplsL3VpnVrfPerfTable	This table specifies per MPLS L3VPN VRF Table performance information
mplsL3VpnVrfRteTable	This table specifies per-interface MPLS L3VPN VRF Table routing information. Entries in this table define VRF routing entries associated with the specified MPLS/VPN interfaces. Note that this table contains both BGP and Interior Gateway Protocol IGP routes, as both may appear in the same VRF

MIB Constraints

This table lists the constraints that the router places on objects in the MPLS-L3VPN-STD-MIB. For detailed definitions of MIB objects, see the MIB.

Table 1-113 MPLS-L3VPN-STD-MIB Constraints

MIB Object	Notes
mplsL3VpnPerfGroup	
mplsL3VpnVrfPerfRoutesAdded	Read-only, set to zero by default.
mplsL3VpnVrfPerfRoutesDeleted	Read-only, set to zero by default.

Table 1-113 MPLS-L3VPN-STD-MIB Constraints

MIB Object	Notes
mplsL3VpnVrfTRteGroup	
mplsL3VpnVrfRteInetCidrNextHopAS	Read-only, set to zero by default.
mplsL3VpnSecGroup	
mplsL3VpnVrfSecIllegalLblVltns	Read-only, set to zero by default.
mplsL3VpnVrfSecDiscontinuityTime	Read-only, set to zero by default.
mplsL3VpnPerfRouteGroup	
mplsL3VpnVrfPerfRoutesDropped	Read-only, set to zero by default.
mplsL3VpnVrfGroup	
mplsL3VpnVrfVpnId	Read-only, set to zero-length OCTET STRING.
mplsL3VpnVrfConfMaxRoutes	Read-only, set to zero by default.
mplsL3VpnScalarGroup	
mplsL3VpnVrfConfMaxPossRts	Read-only, set to zero-length OCTET STRING.
mplsL3VpnIllLblRcvThrsh	Read-only, set to zero by default.

MPLS-LDP-GENERIC-STD-MIB

The MPLS-LDP-GENERIC-STD-MIB contains managed object definitions for configuring and monitoring the Multiprotocol Label Switching (MPLS), Label Distribution Protocol (LDP), utilizing ethernet as the Layer 2 media.

This table lists the tables associated with this MIB.

Table 1-114 MPLS-LDP-GENERIC-STD-MIB Tables and Descriptions

Name	Description
mplsLdpEntityGenericLRTable	MPLS LDP Entity Generic LR Table. The purpose of this table is to provide a mechanism for configurating a contiguous range of generic labels, or a 'label range' for LDP Entities. LDP Entities, which use Generic Labels, must have at least one entry in this table. In other words, this table 'extends' the mpldLdpEntityTable for Generic Labels. There is read-only support for all objects in this table.

MPLS-LDP-STD-MIB

The MPLS-LDP-STD-MIB contains managed object definitions for the ‘Multiprotocol Label Switching, Label Distribution Protocol, LDP document’.

Note Only MANDATORY-GROUPS, which include mplsLdpGeneralGroup and mplsLdpNotificationGroup, are supported.

This table lists the tables associated with this MIB.

Table 1-115 MPLS-LDP-STD-MIB Tables and Descriptions

Name	Description
mplsLdpEntityTable	This table contains information about the MPLS Label Distribution Protocol Entities which exist on this LSR or LER
mplsLdpEntityStatsTable	This table is a read-only table which augments the mplsLdpEntityTable. The purpose of this table is to keep statistical information about the LDP Entities on the LSR
mplsLdpPeerTable	Information about LDP peers known by Entities in the mplsLdpEntityTable. The information in this table is based on information from the Entity-Peer interaction during session initialization but is not appropriate for the mplsLdpSessionTable, because objects in this table may or may not be used in session establishment
mplsLdpSessionTable	Table of Sessions between the LDP Entities and LDP Peers. This table AUGMENTS the mplsLdpPeerTable. Each row in this table represents a single session
mplsLdpSessionStatsTable	Table of statistics for Sessions between LDP Entities and LDP Peers. This table AUGMENTS the mplsLdpPeerTable
mplsLdpHelloAdjacencyTable	Table of Hello Adjacencies for Sessions
mplsFecTable	This table represents the FEC Information associated with an LSP
mplsLdpSessionPeerAddrTable	This table 'extends' the mplsLdpSessionTable. This table is used to store Label Address Information from Label Address Messages received by this LSR from Peers. This table is read-only and should be updated when Label Withdraw Address Messages are received, that is, Rows should be deleted as appropriate. **Note**: Since more than one address may be contained in a Label Address Message, this table 'sparse augments', the mplsLdpSessionTable's information

MPLS-LSR-STD-MIB

The MPLS-LSR-STD-MIB contains managed object definitions for the Multiprotocol Label Switching (MPLS) Router as defined in: Rosen, E., Viswanathan, A., and R. Callon, Multiprotocol Label Switching Architecture, RFC 3031, January 2001.

Note Only MANDATORY-GROUPS which include mplsInterfaceTable, mplsInSegmentTable, mplsOutSegmentTable, mplsXCTable and mplsInterfacePerfTable are supported.

This table lists the tables associated with this MIB.

Table 1-116 MPLS-LSR-STD-MIB Tables and Descriptions

Name	Description
mplsInterfaceTable	This table specifies per-interface MPLS capability and associated information
mplsInterfacePerfTable	This table provides MPLS performance information on a per-interface basis
mplsInSegmentTable	This table contains a description of the incoming MPLS segments (labels) to an LSR and their associated parameters. The index for this table is mplsInSegmentIndex. The index structure of this table is specifically designed to handle many different MPLS implementations that manage their labels both in a distributed and centralized manner. The table is also designed to handle existing MPLS labels as defined in RFC 3031 as well as longer ones that may be necessary in the future. In

Table 1-116 *MPLS-LSR-STD-MIB Tables and Descriptions*

Name	Description
	cases where the label cannot fit into the mplsInSegmentLabel object, the mplsInSegmentLabelPtr indicates this by being set to the first accessible column in the appropriate extension table's row. In this case an additional table MUST be provided and MUST be indexed by at least the indexes used by this table. In all other cases when the label is represented within the mplsInSegmentLabel object, the mplsInSegmentLabelPtr MUST be set to 0.0. Due to the fact that MPLS labels may not exceed 24 bits, the mplsInSegmentLabelPtr object is only a provision for future-proofing the MIB module. Thus, the definition of any extension tables is beyond the scope of this MIB module
mplsInSegmentPerfTable	This table contains statistical information for incoming MPLS segments to an LSR
mplsOutSegmentTable	This table contains a representation of the outgoing segments from an LSR
mplsOutSegmentPerfTable	This table contains statistical information about outgoing segments from an LSR. The counters in this entry should behave in a manner similar to that of the interface
mplsXCTable	This table specifies information for switching between LSP segments. It supports point-to-point, point-to-multipoint and multipoint-to-point connections. mplsLabelStackTable specifies the label stack information for a cross-connect LSR and is referred to from mplsXCTable
mplsLabelStackTable	This table specifies the label stack to be pushed onto a packet, beneath the top label. Entries into this table are referred to from mplsXCTable
mplsInSegmentMapTable	This table specifies the mapping from the mplsInSegmentIndex to the corresponding mplsInSegmentInterface and mplsInSegmentLabel objects. The purpose of this table is to provide the manager with an alternative means by which to locate in-segments

MIB Constraints

This table lists the constraints that the router places on objects in the MPLS-LSR-STD-MIB. For detailed definitions of MIB objects, see the MIB.

Table 1-117 *MPLS-LSR-STD-MIB Constraints*

MIB Object	Notes
mplsSegmentTable	
mplsInSegmentMapIndex	Not supported
mplsLabelStackTable	
mplsLabelStackLabel	Not supported
mplsLabelStackLabelPtr	Not supported
mplsLabelStackRowStatus	Not supported
mplsLabelStackStorageType	Not supported

MPLS-TC-STD-MIB

The MPLS-TC-STD-MIB defines TEXTUAL-CONVENTIONs for concepts used in Multiprotocol Label Switching (MPLS) networks. This MIB has no tables.

MPLS-TE-STD-MIB

The MPLS-TE-STD-MIB contains managed object definitions for the MPLS Traffic Engineering (TE).

This table lists the tables associated with this MIB.

Table 1-118 MPLS-TE-STD-MIB Tables and Descriptions

Name	Description
mplsTunnelTable	mplsTunnelTable allows new MPLS tunnels to be created between an LSR and a remote endpoint, and existing tunnels to be reconfigured or removed. Note that only point-to-point tunnel segments are supported, although multipoint-to-point and point- to-multipoint connections are supported by an LSR acting as a cross-connect. Each MPLS tunnel can have one out-segment originating at this LSR or one in-segment terminating at this LSR
mplsTunnelHopTable	mplsTunnelHopTable is used to indicate the hops, strict or loose, for an instance of an MPLS tunnel defined in mplsTunnelTable, when it is established via signaling, for the outgoing direction of the tunnel. Thus at a transit LSR, this table contains the desired path of the tunnel from this LSR onwards. Each row in this table is indexed by mplsTunnelHopListIndex which corresponds to a group of hop lists or path options. Each row also has a secondary index mplsTunnelHopIndex, which indicates a group of hops (also known as a path option). Finally, the third index, mplsTunnelHopIndex indicates the specific hop information for a path option. To specify a particular interface on the originating LSR of an outgoing tunnel for packets to exit the LSR, specify this as the first hop for this tunnel in mplsTunnelHopTable
mplsTunnelResourceTable	mplsTunnelResourceTable allows a manager to specify which resources are desired for an MPLS tunnel. This table also allows several tunnels to point to a single entry in this table, implying that these tunnels should share resources
mplsTunnelARHopTable	mplsTunnelARHopTable is used to indicate the hops for an MPLS tunnel defined in mplsTunnelTable, as reported by the MPLS signaling protocol. Thus at a transit LSR, this table (if the table is supported and if the signaling protocol is recording actual route information) contains the actual route of the whole tunnel. If the signaling protocol is not recording the actual route, this table MAY report the information from the mplsTunnelHopTable or the mplsTunnelCHopTable. Each row in this table is indexed by mplsTunnelARHopListIndex. Each row also has a secondary index mplsTunnelARHopIndex, corresponding to the next hop that this row corresponds to. Note that since the information necessary to build entries within this table is not provided by some MPLS signaling protocols, implementation of this table is optional. Furthermore, because the information in this table is actually provided by the MPLS signaling protocol after the path has been set-up, the entries in this table are provided only for observation, and hence, all variables in this table are accessible exclusively as read-only. Note also that the contents of this table may change while it is being read because

Table 1-118 MPLS-TE-STD-MIB Tables and Descriptions

Name	**Description**
	of re-routing activities. A network administrator may verify that the actual route read is consistent by reference to the mplsTunnelLastPathChange object
mplsTunnelCHopTable	mplsTunnelCHopTable is used to indicate the hops, strict or loose, for an MPLS tunnel defined in mplsTunnelTable, as computed by a constraint- based routing protocol, based on the mplsTunnelHopTable for the outgoing direction of the tunnel. Thus at a transit LSR, this table (if the table is supported) MAY contain the path computed by the CSPF engine on (or on behalf of) this LSR. Each row in this table is indexed by mplsTunnelCHopListIndex. Each row also has a secondary index mplsTunnelCHopIndex, corresponding to the next hop that this row corresponds to. In case we want to specify a particular interface on the originating LSR of an outgoing tunnel by which we want packets to exit the LSR, we specify this as the first hop for this tunnel in mplsTunnelCHopTable
mplsTunnelPerfTable	This table provides per-tunnel instance MPLS performance information
mplsTunnelCRLDPResTable	mplsTunnelCRLDPResTable allows a manager to specify which CR-LDP-specific resources are desired for an MPLS tunnel if that tunnel is signaled using CR-LDP. Note that these attributes are in addition to those specified in mplsTunnelResourceTable. This table also allows several tunnels to point to a single entry in this table, implying that these tunnels should share resources

MIB Constraints

This table lists the constraints on objects in the MPLS-TE-STD-MIB.

Table 1-119 MPLS-TE-STD-MIB Constraints

MIB Object	**Notes**
mplsTunnelCRLDPResTable	CRLDP signaling not supported for Traffic Engineering.

NOTIFICATION-LOG-MIB

The NOTIFICATION-LOG-MIB is for logging SNMP Notifications, that is, Traps and Informs.

This table lists the tables associated with this MIB.

Table 1-120 NOTIFICATION-LOG-MIB Tables and Descriptions

Name	**Description**
nlmConfigLogTable	Table of logging control entries
nlmStatsLogTable	Table of Notification log statistics entries
nlmLogTable	Table of Notification log entries. It is an implementation-specific matter whether entries in this table are preserved across initializations of the management system. In general, one would expect that they are not. Note that keeping entries across initializations of the management system leads to some confusion with counters and TimeStamps, since both of those are based on sysUpTime, which resets on management

Table 1-120 NOTIFICATION-LOG-MIB Tables and Descriptions

Name	Description
	initialization. In this situation, counters apply only after the reset and nlmLogTime for entries made before the reset must be set to 0
nlmLogVariableTable	Table of variables to go with Notification log entries

OSPF-MIB

The OSPF-MIB module describes the OSPF Version 2 Protocol. Note that some objects in this MIB module may pose a significant security risk. See the Security Considerations section in RFC 4750 for more information.

This table lists the tables associated with this MIB.

Table 1-121 OSPF-MIB Tables and Descriptions

Name	Description
ospfAreaTable	Information describing the configured parameters and cumulative statistics of the router's attached areas. The interfaces and virtual links are configured as part of these areas. Area 0.0.0.0, by definition, is the backbone area
ospfStubAreaTable	Set of metrics that is advertised by a default Area Border Router into a stub area
ospfLsdbTable	OSPF Process'sLSDB[1]. The LSDB contains the link state advertisements from throughout the areas that the device is attached to
ospfAreaRangeTable	Address Range Table acts as an adjunct to the Area Table. It describes those Address Range Summaries that are configured to be propagated from an Area to reduce the amount of information about it that is known beyond its borders. It contains a set of IP address ranges specified by an IP address/IP network mask pair. For example, class B address range of x.x.x.x with a network mask of 255.255.0.0 includes all IP addresses from x.x.0.0 to x.x.255.255. Note that this table is obsoleted and is replaced by the Area Aggregate Table
ospfHostTable	Host/Metric Table indicates what hosts are directly attached to the router, what metrics and types of service should be advertised for them, and what areas they are found within
ospfIfTable	OSPF Interface Table describes the interfaces from the viewpoint of OSPF. It augments the ipAddrTable with OSPF specific information
ospfIfMetricTable	Metric Table describes the metrics to be advertised for a specified interface at the various types of service. As such, this table is an adjunct of the OSPF Interface Table. Types of service, as defined by RFC 791, have the ability to request low delay, high bandwidth, or reliable linkage. For the purposes of this specification, the measure of bandwidth: Metric = referenceBandwidth / ifSpeed is the default value. The default reference bandwidth is 10^8. For multiple link interfaces, note that ifSpeed is the sum of the individual link speeds. This yields a number having the following typical values: Network Type/bit rate Metric >= 100 MBPS 1

Table 1-121 *OSPF-MIB Tables and Descriptions*

Name	Description
	Ethernet/802.3 10 E1 48 T1 (ESF) 65 64 KBPS 1562 56 KBPS 1785 19.2 KBPS 5208 9.6 KBPS 10416 Routes that are not specified use the default (TOS 0) metric. Note that the default reference bandwidth can be configured using the general group object ospfReferenceBandwidth.
ospfVirtIfTable	Information about this router's virtual interfaces that the OSPF Process is configured to carry on
ospfNbrTable	Table describing all non-virtual neighbors in the locality of the OSPF router
ospfVirtNbrTable	This table describes all virtual neighbors. Since virtual links are configured in the Virtual Interface Table, this table is read-only
ospfExtLsdbTable	OSPF Process's external LSA link state database. This table is identical to the OSPF LSDB Table in format, but contains only external link state advertisements. The purpose is to allow external LSAs to be displayed once for the router rather than once in each non-stub area. Note that external LSAs are also in the AS-scope link state database
ospfAreaAggregateTable	Area Aggregate Table acts as an adjunct to the Area Table. It describes those address aggregates that are configured to be propagated from an area. Its purpose is to reduce the amount of information that is known beyond an Area's borders. It contains a set of IP address ranges specified by an IP address/IP network mask pair. For example, a class B address range of x.x.x.x with a network mask of 255.255.0.0 includes all IP addresses from x.x.0.0 to x.x.255.255. **Note** If ranges are configured such that one range subsumes another range (that is, 10.0.0.0 mask 255.0.0.0 and 10.1.0.0 mask 255.255.0.0), the most specific match is the preferred one. See OSPF Version 2, Appendix C.2 Area parameters
ospfLocalLsdbTable	OSPF Process's link-local link state database for non-virtual links. This table is identical to the OSPF LSDB Table in format, but contains only link-local Link State Advertisements for non-virtual links. The purpose is to allow link-local LSAs to be displayed for each non-virtual interface. This table is implemented to support type-9 LSAs that are defined in OSPF Opaque LSA Option.
ospfVirtLocalLsdbTable	OSPF Process's link-local link state database for virtual links. This table is identical to the OSPF LSDB Table in format, but contains only link-local Link State Advertisements for virtual links. The purpose is to allow link-local LSAs to be displayed for each virtual interface. This table is implemented to support type-9 LSAs that are defined in OSPF Opaque LSA Option.
ospfAsLsdbTable	OSPF Process's AS-scope LSA link state database. The database

Table 1-121 ***OSPF-MIB Tables and Descriptions***

Name	Description
	contains the AS-scope Link State Advertisements from throughout the areas that the device is attached to. This table is identical to the OSPF LSDB Table in format, but contains only AS-scope Link State Advertisements. The purpose is to allow AS-scope LSAs to be displayed once for the router rather than once in each non-stub area
ospfAreaLsaCountTable	This table maintains per-area, per-LSA-type counters

1. LSDB = link state database

OSPF-TRAP-MIB

The OSPF-TRAP-MIB describes the traps for OSPF Version 2 Protocol. This MIB has no tables.

OSPFV3-MIB

The OSPFV3-MIB is the MIB module for OSPF version 3.

This table lists the tables associated with this MIB.

Table 1-122 ***OSPFV3-MIB Tables and Descriptions***

Name	Description
ospfv3AreaTable	OSPFv3 Process's AS-Scope LSDB. The LSDB contains the AS-Scope Link State Advertisements from throughout the areas that the device is attached to.
ospfv3AsLsdbTable	OSPFv3 Process's AS-Scope LSDB. The LSDB contains the AS-Scope Link State Advertisements from throughout the areas that the device is attached to.
ospfv3AreaLsdbTable	OSPFv3 Process's Area-Scope LSDB. The LSDB contains the Area-Scope Link State Advertisements from throughout the area that the device is attached to.
ospfv3LinkLsdbTable	OSPFv3 Process's Link-Scope LSDB for non-virtual interfaces. The LSDB contains the Link-Scope Link State Advertisements from the interfaces that the device is attached to
ospfv3HostTable	Host/Metric Table indicates what hosts are directly attached to the router and their corresponding metrics
ospfv3IfTable	OSPFv3 Interface Table describes the interfaces from the viewpoint of OSPFv3
ospfv3VirtIfTable	Information about this router's virtual interfaces that the OSPFv3 Process is configured to carry on
ospfv3NbrTable	A table describing all neighbors in the locality of the OSPFv3 router
ospfv3CfgNbrTable	Table describing all configured neighbors
ospfv3VirtNbrTable	Table describing all virtual neighbors
ospfv3AreaAggregateTable	Area Aggregate Table acts as an adjunct to the Area Table. It describes those address aggregates that are configured to be propagated from an area. Its purpose is to reduce the amount of information that is known beyond an Area's borders. A range of IPv6 prefixes specified by a prefix/prefix length pair. Note that if ranges are configured such that one range subsumes another range the most specific match is the preferred

Table 1-122 OSPFV3-MIB Tables and Descriptions

Name	Description
	one
ospfv3VirtLinkLsdbTable	OSPFv3 Process's Link-Scope LSDB for virtual interfaces. The LSDB contains the Link-Scope Link State Advertisements from virtual interfaces

RADIUS-ACC-CLIENT-MIB

The RADIUS-ACC-CLIENT-MIB is the MIB module for entities implementing the client side of the RADIUS accounting protocol.

This table lists the tables associated with this MIB.

Table 1-123 RADIUS-ACC-CLIENT-MIB Tables and Descriptions

Name	Description
radiusAccServerTable	(conceptual) Table listing the RADIUS accounting servers with which the client shares a secret.

RADIUS-AUTH-CLIENT-MIB

The RADIUS-AUTH-CLIENT-MIB is the MIB module for entities implementing the client side of the RADIUS authentication protocol.

This table lists the tables associated with this MIB.

Table 1-124 RADIUS-AUTH-CLIENT-MIB Tables and Descriptions

Name	Description
radiusAuthServerTable	(conceptual) Table listing the RADIUS authentication servers with which the client shares a secret.

RFC 1213-MIB

The RFC 1213-MIB is the second version of the MIB-II for using with network management protocols in TCP-based networks. This MIB is superseded by separate standard MIBs for MIB II (UDP, TCP, IP and so forth).

Note For more information on the evolution of RFC 1213-MIB see Appendix C, “RFC 1213”.

This table lists the tables associated with this MIB.

Table 1-125 RFC 1213-MIB Tables and Descriptions

Name	Description
ifTable	List of interface entries. The number of entries is given by the value of ifNumber.
atTable	Address Translation tables contain the NetworkAddress to `physical' address equivalences. Some interfaces do not use translation tables for determining address equivalences (for example, DDN-X.25 has an algorithmic method); if all interfaces are of this type, then the Address Translation table is empty, in other words, has zero entries.

Table 1-125 ***RFC 1213-MIB Tables and Descriptions***

Name	Description
ipAddrTable	Table of addressing information relevant to this entity's IP addresses.
ipRouteTable	This entity's IP Routing table.
ipNetToMediaTable	IP Address Translation table used for mapping from IP addresses to physical addresses.
tcpConnTable	Table containing TCP connection-specific information.
udpTable	Table containing UDP listener information.
egpNeighTable	EGP neighbor table.

RSVP-MIB

The RSVP-MIB contains the tables which contain RSVP specific information. RSVP is a signaling protocol that enables systems to request resource reservations from the network. RSVP processes protocol messages from other systems, processes resource requests from local clients, and generates protocol messages. As a result, resources are reserved for data flows on behalf of local and remote clients. RSVP creates, maintains and deletes these resource reservations and the MIB provides status data corresponding to this.

This table lists the tables associated with this MIB.

Table 1-126 ***RSVP-MIB Tables and Descriptions***

Name	Description
ifTable	List of interface entries. The number of entries is given by the value of ifNumber.
atTable	Address Translation tables contain theNetwork Address to physical address equivalences. Some interfaces do not use translation tables for determining address equivalences (that is, DDN-X.25 has an algorithmic method); if all interfaces are of this type, then the Address Translation table is empty, that is, has zero entries.
ipAddrTable	Table of addressing information relevant to this entity's IP addresses.
ipRouteTable	This entity's IP Routing table.
ipNetToMediaTable	IP Address Translation table used for mapping from IP addresses to physical addresses.
tcpConnTable	Table containing TCP connection-specific information.
udpTable	Table containing UDP listener information.
egpNeighTable	EGP neighbor table.

MIB Constraints

This table lists the constraints on objects in the RSVP-MIB.

Table 1-127 ***RSVP-MIB Constraints***

MIB Object	Notes
rsvpResvFwdNewIndex	Not supported

Table 1-127 RSVP-MIB Constraints

MIB Object	**Notes**
rsvpSenderNewIndex	Not supported
rsvpBadPackets	Not supported
rsvpResvNewIndex	Not supported
rsvpSession NewIndex	Not supported

SNMP-COMMUNITY-MIB (RFC 2576)

The SNMP-COMMUNITY-MIB (RFC 2576) contains objects that help support coexistence between SNMPv1, SNMPv2c, and SNMPv3.

Note For more information on SNMP-COMMUNITY-MIB see Appendix D, “Process Information for SNMP-centric MIBs”.

This table lists the tables associated with this MIB.

Table 1-128 SNMP-COMMUNITY-MIB Tables and Descriptions

Name	**Description**
snmpCommunityTable	Table of community strings configured in the SNMP engine's LCD.
snmpTargetAddrExtTable	Table of mask and mms values associated with the snmpTargetAddrTable. The snmpTargetAddrExtTable augments the snmpTargetAddrTable with a transport address mask value and a maximum message size value. The transport address mask allows entries in the snmpTargetAddrTable to define a set of addresses instead of just a single address. The maximum message size value allows the maximum message size of another SNMP entity to be configured for use in SNMPv1 (and SNMPv2c) transactions, where the message format does not specify a maximum message size.

SNMP-FRAMEWORK-MIB (RFC 2571)

The SNMP-FRAMEWORK-MIB (RFC 2571) contains objects that describe the SNMP management architecture. There are no constraints on this MIB. This MIB has no tables.

Note For more information on SNMP-centric MIBs see Appendix D, “Process Information for SNMP-centric MIBs”.

SNMP-MPD-MIB (RFC 2572)

The SNMP-MPD-MIB is the MIB for message processing and dispatching. This MIB has no tables.

Note For more information on SNMP-centric MIBs see Appendix D, “Process Information for SNMP-centric MIBs”.

SNMP-NOTIFICATION-MIB (RFC 2573)

The SNMP-NOTIFICATION-MIB contains managed objects for SNMPv3 notifications. The MIB also defines a set of filters that limit the number of notifications generated by a particular entity (snmpNotifyFilterProfileTable and snmpNotifyFilterTable).

Objects in the snmpNotifyTable are used to select entities in the SNMP-TARGET-MIB snmpTargetAddrTable and specify the types of supported SNMP notifications.

Note For more information on SNMP-centric MIBs see Appendix D, "Process Information for SNMP-centric MIBs".

This table lists the tables associated with this MIB.

Table 1-129 ***SNMP-NOTIFICATION-MIB Tables and Descriptions***

Name	**Description**
snmpNotifyTable	This table is used to select management targets which should receive notifications, as well as the type of notification which should be sent to each selected management target
snmpNotifyFilterProfileTable	This table is used to associate a notification filter profile with a particular set of target parameters
snmpNotifyFilterTable	Table of filter profiles. Filter profiles are used to determine whether particular management targets should receive particular notifications. When a notification is generated, it must be compared with the filters associated with each management target which is configured to receive notifications, to determine whether it may be sent to each such management target. A more complete discussion of notification filtering can be found in (RFC 2573)

SNMP-TARGET-MIB (RFC 2573)

The SNMP-TARGET-MIB (RFC 2573) contains objects to remotely configure the parameters used by an entity to generate SNMP notifications. The MIB defines the addresses of the destination entities for SNMP notifications and contains a list of tag values that are used to filter the notifications sent to the entities (see the SNMP-NOTIFICATION-MIB). There are no constraints on this MIB.

Note For more information on SNMP-centric MIBs see Appendix D, "Process Information for SNMP-centric MIBs".

This table lists the tables associated with this MIB.

Table 1-130 ***SNMP-TARGET-MIB Tables and Descriptions***

Name	**Description**
snmpTargetAddrTable	Table of transport addresses to be used in the generation of SNMP messages
snmpTargetParamsTable	Table of SNMP target information to be used in the generation of SNMP messages

SNMP-USM-MIB (RFC 2574)

The SNMP-USM-MIB (RFC 2574) contains objects that describe the SNMP user-based security model.

Note For more information on SNMP-USM-MIB see Appendix D, “Process Information for SNMP-centric MIBs”.

This table lists the tables associated with this MIB.

Table 1-131 SNMP-USM-MIB Tables and Descriptions

Name	Description
usmUserTable	Table of users configured in the SNMP engine's LCD. To create a new user (that is, to instantiate a new conceptual row in this table), it is recommended to follow this procedure: 1) GET(usmUserSpinLock.0) and save in sValue. 2) SET(usmUserSpinLock.0=sValue, usmUserCloneFrom=templateUser, usmUserStatus=createAndWait) You should use a template user to clone from which has the proper auth/priv protocol defined. If the new user is to use privacy: 3) Generate the keyChange value based on the secret privKey of the clone-from user and the secret key to be used for the new user. Let us call this pkcValue. 4) GET(usmUserSpinLock.0) and save in sValue. 5) SET(usmUserSpinLock.0=sValue, smUserPrivKeyChange=pkcValue usmUserPublic=randomValue1) 6) GET(usmUserPulic) and check it has randomValue1. If not, repeat steps 4-6. If the new user will never use privacy: 7) SET(usmUserPrivProtocol=usmNoPrivProtocol) If the new user is to use authentication: 8) Generate the keyChange value based on the secret authKey of the clone-from user and the secret key to be used for the new user. Let us call this akcValue. 9)GET(usmUserSpinLock.0) and save in sValue. 10) SET(usmUserSpinLock.0=sValue, usmUserAuthKeyChange=akcValueusmUserPublic=randomValue2) 11) GET(usmUserPulic) and check it has randomValue2. If not, repeat steps 9 to11. If the new user will never use authentication: 12) SET(usmUserAuthProtocol=usmNoAuthProtocol) Finally, activate the new user: 13) SET(usmUserStatus=active)The new user should now be available and ready to be used for SNMPv3communication. Note however that access to MIB data must be provided viaconfiguration of the SNMP-VIEW-BASED-ACM-MIB. The use ofusmUser Spinlock is to avoid conflicts with another SNMP command responderapplication which may also be acting on the usmUserTable

SNMP-VACM-MIB (RFC 2575)

The SNMP-VACM-MIB contains objects to manage the View-Based Access Control Model (VACM) for SNMP clients and managers. The MODULE-IDENTITY for the SNMP-VACM-MIB is snmpVacmMIB, and its top-level OID is 1.3.6.1.6.3.16 (iso.org.dod.internet.snmpv2.snmpModules.snmpVacmMIB).

Note For more information on SNMP-VACM-MIB see Appendix D, “Process Information for SNMP-centric MIBs”.

This table lists the tables associated with this MIB.

Table 1-132 SNMP-VACM-MIB Tables and Descriptions

Name	Description
vacmContextTable	Table of locally available contexts. This table provides information to SNMP Command Generator applications so that they can properly configure the vacmAccessTable to control access to all contexts at the SNMP entity. This table may change dynamically if the SNMP entity allows that contexts are added or deleted dynamically (for instance when its configuration changes). Such changes would happen only if the management instrumentation at that SNMP entity recognizes more (or fewer) contexts. The presence of entries in this table and of entries in the vacmAccessTable are independent. That is, a context identified by an entry in this table is not necessarily referenced by any entries in the vacmAccessTable; and the context(s) referenced by an entry in the vacmAccessTable does not necessarily currently exist and thus need not be identified by an entry in this table. This table must be made accessible via the default context so that Command Responder applications have a standard way of retrieving the information. This table is read-only. It cannot be configured via SNMP
vacmSecurityToGroupTable	This table maps a combination of securityModel and securityName into a groupName which is used to define an access control policy for a group of principals
vacmAccessTable	Table of access rights for groups. Each entry is indexed by a groupName, a contextPrefix, a securityModel and a securityLevel. To determine whether access is allowed, one entry from this table needs to be selected and the proper viewName from that entry must be used for access control checking. To select the proper entry, follow these steps: 1) The set of possible matches is formed by the intersection of the following sets of entries: the set of entries with identical vacmGroupName the union of these two sets: ■ Set with identical vacmAccessContextPrefix ■ Set of entries with vacmAccessContextMatch value of 'prefix' and matching vacmAccessContextPrefix intersected with the union of these two sets: ■ Set of entries with identical vacmSecurityModel ■ Set of entries with vacmSecurityModel value of 'any' intersected with the set of entries with vacmAccessSecurityLevel value less than or equal to the requested securityLevel 2) If this set has only one member, we are finished, otherwise, it comes down to deciding how to weight the preferences between ContextPrefixes, SecurityModels, and SecurityLevels as follows: a) If the subset of entries with securityModel matching the securityModel in the message is not empty, then discard the rest. b) If the subset of entries with vacmAccessContextPrefix matching the contextName in the message is not empty, then discard the rest c) Discard all entries with ContextPrefixes shorter than the longest one remaining in the set d) Select the entry with the highest securityLevel Note that for securityLevel noAuthNoPriv, all groups are really equivalent since the assumption that the securityName has been authenticated does not hold

Table 1-132 ***SNMP-VACM-MIB Tables and Descriptions***

Name	Description
vacmViewTreeFamily-Table	Locally held information about families of subtrees within MIB views. Each MIB view is defined by two sets of view subtrees: - Included view subtrees - Excluded view subtrees. Every such view subtree, both the included and the excluded ones, is defined in this table. To determine if a particular object instance is in a particular MIB view, compare the object instance's OBJECT IDENTIFIER with each of the MIB view's active entries in this table. If none match, then the object instance is not in the MIB view. If one or more match, then the object instance is included in, or excluded from, the MIB view according to the value of vacmViewTreeFamilyType in the entry whose value of vacmViewTreeFamilySubtree has the most sub-identifiers. If multiple entries match and have the same number of sub-identifiers (when wildcarding is specified with the value of vacmViewTreeFamilyMask), then the lexicographically greatest instance of vacmViewTreeFamilyType determines the inclusion or exclusion. An object instance's OBJECT IDENTIFIER X matches an active entry in this table when the number of sub-identifiers in X is at least as many as in the value of vacmViewTreeFamilySubtree for the entry, and each sub-identifier in the value of vacmViewTreeFamilySubtree matches its corresponding sub-identifier in X. Two sub-identifiers match either if the corresponding bit of the value of vacmViewTreeFamilyMask for the entry is zero (the 'wild card' value), or if they are equal. A 'family' of subtrees is the set of subtrees defined by a particular combination of values of vacmViewTreeFamilySubtree and vacmViewTreeFamilyMask. In the case where no 'wild card' is defined in the vacmViewTreeFamilyMask, the family of subtrees reduces to a single subtree. When creating or changing MIB views, an SNMP Command Generator application should utilize the vacmViewSpinLock to try to avoid collisions. See DESCRIPTION clause of vacmViewSpinLock. When creating MIB views, it is strongly advised that first the 'excluded' vacmViewTreeFamilyEntries are created and then the 'included' entries. When deleting MIB views, it is strongly advised that first the 'included' vacmViewTreeFamilyEntries are deleted and then the 'excluded' entries. If a create for an entry for instance-level access control is received and the implementation does not support instance-level granularity, then an inconsistentName error must be returned.

SNMPv2-MIB (RFC 1907)

The SNMPv2-MIB contains objects SNMPv2 entities. The SNMPv2-MIB contains the following mandatory object groups:

- SNMP group–Collection of objects providing basic instrumentation and control of an SNMP entity.
- System group–Collection of objects common to all managed systems.

- snmpSetGroup–Collection of objects that allow several cooperating SNMPv2 entities, all acting in a manager role, to coordinate their use of the SNMPv2 set operation.
- snmpBasicNotificationsGroup–Two notifications are coldStart and authenticationFailure, which an SNMPv2 entity is required to implement.

Note For more information on SNMP-centric MIBs see Appendix D, “Process Information for SNMP-centric MIBs”.

This table lists the tables associated with this MIB.

Table 1-133 SNMPv2-MIB Tables and Descriptions

Name	Description
sysORTable	(conceptual) Table listing the capabilities of the local SNMP application acting as a command responder about various MIB modules. SNMP entities having dynamically-configurable support of MIB modules have a dynamically-varying number of conceptual rows

SONET-MIB

The SONET-MIB describes SONET/SDH interface objects.

This table lists the tables associated with this MIB.

Table 1-134 SONET-MIB Tables and Descriptions

Name	Description
sonetMediumTable	SONET/SDH Medium table.
sonetSectionCurrentTable	SONET/SDH Section Current table.
sonetSectionIntervalTable	SONET/SDH Section Interval table.
sonetLineCurrentTable	SONET/SDH Line Current table.
sonetLineIntervalTable	SONET/SDH Line Interval table.
sonetFarEndLineCurrentTable	SONET/SDH Far End Line Current table.
sonetFarEndLineIntervalTable	SONET/SDH Far End Line Interval table.
sonetPathCurrentTable	SONET/SDH Path Current table.
sonetPathIntervalTable	SONET/SDH Path Interval table.
sonetFarEndPathCurrentTable	SONET/SDH Far End Path Current table.
sonetFarEndPathIntervalTable	SONET/SDH Far End Path Interval table.
sonetVTCurrentTable	SONET/SDH VT Current table.
sonetVTIntervalTable	SONET/SDH VT Interval table.
sonetFarEndVTCurrentTable	SONET/SDH Far End VT Current table.
sonetFarEndVTIntervalTable	SONET/SDH Far End VT Interval table.

TCP-MIB

The TCP-MIB is the MIB module for managing TCP implementations.

This table lists the tables associated with this MIB.

Table 1-135 TCP-MIB Tables and Descriptions

Name	Description
tcpConnectionTable	Table containing information about existing TCP connections. Note that unlike earlier TCP MIBs, there is a separate table for connections in the LISTEN state
tcpListenerTable	Table containing information about TCP listeners. A listening application can be represented in three possible ways: 1. An application that is willing to accept both IPv4 and IPv6 datagrams is represented by a tcpListenerLocalAddressType of unknown (0) and a tcpListenerLocalAddress of ‘h’ (a zero-length octet-string). 2. An application that is willing to accept only IPv4 or IPv6 datagrams is represented by a tcpListenerLocalAddressType of the appropriate address type and a tcpListenerLocalAddress f '0.0.0.0' or '::' respectively. 3. An application that is listening for data destined only to a specific IP address, but from any remote system, is represented by a tcpListenerLocalAddressType of an appropriate address type, with tcpListenerLocalAddress as the specific local address. **Note** The address type in this table represents the address type used for the communication, irrespective of the higher-layer abstraction. For example, an application using IPv6 'sockets' to communicate via IPv4 between ::ffff:10.0.0.1 and ::ffff:10.0.0.2 would use InetAddressType IPv4(1))
tcpConnTable	Table containing information about existing IPv4-specific TCP connections or listeners. This table has been deprecated in favor of the version neutral tcpConnectionTable

MIB Constraints

This table lists the constraints that the router places on objects in the TCP-MIB.

Table 1-136 TCP-MIB Constraints

MIB Object	Notes
tcpConnectionTable	
tcpConnectionProcess	Not supported

UDP-MIB

The UDP-MIB is the MIB module for UDP implementations. See RFC 4113.

This table lists the tables associated with this MIB.

Table 1-137 UDP-MIB Tables and Descriptions

Name	Description
udpEndpointTable	Table containing information about this entity's UDP endpoints on which a local application is currently accepting or sending datagrams. The address type in this table represents the address type used for the communication, irrespective of the higher-layer abstraction. For

Table 1-137 *UDP-MIB Tables and Descriptions*

Name	Description
	example, an application using IPv6 'sockets' to communicate via IPv4 between ::ffff:10.0.0.1 and ::ffff:10.0.0.2 would use InetAddressType IPv4(1). Unlike the udpTable in RFC 2013, this table also allows the representation of an application that completely specifies both local and remote addresses and ports. A listening application is represented in three possible ways: 1) An application that is willing to accept both IPv4 and IPv6 datagrams is represented by a udpEndpointLocalAddressType of unknown(0) and a udpEndpointLocalAddress of h (a zero-length octet-string). 2) An application that is willing to accept only IPv4 or only IPv6 datagrams is represented by a udpEndpointLocalAddressType of the appropriate address type and a udpEndpointLocalAddress of '0.0.0.0' or '::' respectively. 3) An application that is listening for datagrams only for a specific IP address but from any remote system is represented by a udpEndpointLocalAddressType of the appropriate address type, with udpEndpointLocalAddress specifying the local address. In all cases where the remote is a wildcard, the udpEndpointRemoteAddressType is unknown(0), the udpEndpointRemoteAddress is h (a zero-length octet-string), and the udpEndpointRemotePort is 0. If the operating system is demultiplexing UDP packets by remote address and port, or if the application has 'connected' the socket specifying a default remote address and port, the udpEndpointRemote* values should be used to reflect this
udpTable	Table containing IPv4-specific UDP listener information. It contains information about all local IPv4 UDP end-points on which an application is currently accepting datagrams. This table has been replaced by the version neutral udpEndpointTable but is currently still supported on IOS XR.

MIB Constraints

This table lists the constraints that the router places on objects in the UDP-MIB. For detailed definitions of MIB objects, see the MIB.

Table 1-138 *UDP-MIB Constraints*

MIB Object	Notes
tcpConnectionTable	
udpEndPointProcess	Not supported

VPN-TC-STD-MIB

The VPN-TC-STD-MIB contains TCs for VPNs. There are no tables associated with this MIB.

VRRP-MIB

The VRRP-MIB describes objects used for managing Virtual Router Redundancy Protocol (VRRP) routers.

This table lists the tables associated with this MIB.

Table 1-139 VRRP-MIB Tables and Descriptions

Name	Description
vrrpOperTable	Operations table for a VRRP router which consists of a sequence (that is, one or more conceptual rows) of 'vrrpOperEntry' items.
vrrpAssoIpAddrTable	Table of addresses associated with this virtual router.
vrrpRouterStatsTable	Table of virtual router statistics.

Cisco Carrier Routing System MIB Specifications

This chapter describes the Management Information Base (MIB) on the Cisco Carrier Routing System. Each MIB description lists any constraints on how the MIB or its object identifiers (OIDs) are implemented on the Cisco Carrier Routing System.

Unless noted otherwise, the Cisco Carrier Routing System implementation of a MIB follows the standard MIB that has been defined. Any MIB table or object not listed in the table is implemented as defined in the standard MIB definition.

This chapter contains the following sections:

- Cisco Carrier Routing System MIBs
- Cisco Carrier Routing System MIB Categories
- MIB Version String Description
- MIBs in the Cisco Carrier Routing System

Cisco Carrier Routing System MIBs

Each MIB description lists relevant constraints about the implementation of the MIB on the Cisco Carrier Routing System. Any objects not listed in a table are implemented as defined in the MIB. For detailed MIB descriptions, see the standard MIB.

Note Not all MIBs included in a Cisco IOS XR Software release are fully supported by the router. Some MIBs are not supported at all. Other MIBs might work, but they have not been tested on the router. In addition, some MIBs are deprecated but cannot be removed from the software. When a MIB is included in the image, this does not necessarily mean it is supported by the Cisco Carrier Routing System platform.

To determine which MIBs are included in other releases, see the “Downloading and Compiling MIBs” section.

Cisco Carrier Routing System MIB Categories

The MIBs in the Cisco Carrier Routing System are categorized into three types:

- see the “Supported and Verified MIBs” section
- see the “Supported and Unverified MIBs” section
- see the “Unverified or Unsupported MIBs” section

Supported and Verified MIBs

The MIB exists in the image, the code is implemented, and Cisco has verified that all the supported objects work properly. These MIBs are tested for the Cisco Carrier Routing System.

Supported and Unverified MIBs

The MIB exists in the image, the code is implemented, but we have not verified if it is working properly. In other words, the user may get something if they query the MIB. However, the information may be correct or incorrect the MIB has not been tested. These MIBs are not tested for the Cisco Carrier Routing System supports.

Unverified or Unsupported MIBs

The MIB exists in the image but is either not tested or not supported. These MIBs are neither tested nor supported for the Cisco Carrier Routing System.

MIB Version String Description

The MIB version string indicates the date and time that the module was most recently modified. The format is YYMMDDHHMMZ or YYYYMMDDHHMMZ, where:

- YY is the last two digits of the year (only years between 1900 and 1999).
- YYYY is all four digits of the year (any year).
- MM is the month (01 through 12).
- DD is the day of the month (01 through 31).
- HH is hours (00 through 23).
- MM is minutes (00 through 59).
- Z (the ASCII character Z) denotes Coordinated Universal Time (UTC, formerly Greenwich Mean Time, GMT). This datatype stores the date and time fields YEAR, MONTH, DAY, HOUR, MINUTE, SECOND, TIMEZONE_HOUR, and TIMEZONE_MINUTE.

Note For example, 9502192015Z and 199502192015Z represent 8:15 GMT on 19 February 1995. Years after 1999 use the four-digit format. Years 1900–1999 may use the two or four digit format.

Note In the following table the term *Revision not available* refers to the MIB module that does not have a recorded time stamp indicating the latest modification.

MIBs in the Cisco Carrier Routing System

This table lists the MIBs in the Cisco Carrier Routing System.

Table 1-1 MIBs in the Cisco Carrier Routing System

MIB	**midb process**	**Supported**		**Unsupported or Unverified**	**Not in Image**
		Verified	**Unverified**		
CISCO-MAU-EXT-MIB	mibd-interface				
■ Release 3.7		200803050000Z	–	–	–
■ Release 3.9		200803050000Z			
■ Release 4.0		200803050000Z			
■ Release 4.2		200803050000Z			
■ Release 4.3		200803050000Z			
IANA-MAU-MIB	–				
■ Release 3.7		–	–	–	–
■ Release 3.9					
■ Release 4.0					
■ Release 4.2					
■ Release 4.3					
MAU-MIB	mibd-interface				
■ Release 3.7		200309190000Z	–	–	–
■ Release 3.9		200309190000Z			
■ Release 4.0		200309190000Z			
■ Release 4.2		200309190000Z			
■ Release 4.3		200309190000Z			

CISCO-MAU-EXT-MIB

The CISCO-MAU-EXT-MIB extends the MAU-MIB (RFC 3636) to add objects which provide additional management information about MAU or Jack not available in MAU-MIB.

This table lists the tables associated with this MIB.

Table 1-2 CISCO-MAU-EXT-MIB Tables and Descriptions

Name	Description
cmExtJackConfigTable	This table contains management information about MAUs attached to interfaces with managed jacks. The objects in this table are in addition to the objects in the ifJackTable.
cmExtIfAutoMdixConfigTable	This table contains Auto-MDIX configuration information for MAUs attached to interfaces in the device. An entry appears in this table for each active Auto-MDIX capable MAU connected to interfaces in the device.
cmExtIfMauTrafficTable	(conceptual) Table that identifies the type of traffic carried by the interfaces associated with the MAUs in the device. This table is sparse dependant on ifMauTable.

MIB Constraints

This table lists the constraints that the Cisco Carrier Routing System places on objects in the MAU-MIB. For detailed definitions of MIB objects, see the MIB.

Table 1-3 CISCO-MAU-EXT-MIB Constraints

MIB Object	Notes
cmExtJackConfigTable	Not supported
cmExtIfAutoMdixConfigTable	Not supported

IANA-MAU-MIB

The IANA-MAU-MIB does not define any management objects. Instead, it defines a set of textual conventions which are used by the MAU-MIB and may be used by other MIB modules to define management objects. Meaningful security considerations can only be written for MIB modules that define management objects. This MIB is required by the MAU-MIB to export the IANAifMauTypeListBits, IANAifMauMediaAvailable, IANAifMauAutoNegCapBits, and IANAifJackType definitions.

Note None of the objects defined in this MIB are supported.

MAU-MIB

The MAU-MIB describes Management Information for 802.3 MAUs.

This table lists the tables associated with this MIB.

Table 1-4 MAU-MIB Tables and Descriptions

Name	Description
rpMauTable	Table of descriptive and status information about the MAU attached to the ports of a repeater.

Table 1-4 MAU-MIB Tables and Descriptions

Name	Description
rpJackTable	Information about the external jacks attached to MAUs attached to the ports of a repeater.
ifMauTable	Table of descriptive and status information about MAU(s) attached to an interface.
ifJackTable	Information about the external jacks attached to MAUs attached to an interface.
ifMauAutoNegTable	Configuration and status objects for the auto-negotiation function of MAUs attached to interfaces.
broadMauBasicTable	Table of descriptive and status information about the broadband MAUs connected to interfaces. This entire table has been deprecated. There have been no reported implementations of this table, and it is unlikely that there ever will be. IEEE recommends that broadband MAU types should not be used for new installations.

MIB Constraints

This table lists the constraints that the Cisco Carrier Routing System places on objects in the MAU-MIB. For detailed definitions of MIB objects, see the MIB. In addition to the constraints listed in the table, MAU-MIB only supports managing Control-Ethernet (CE) interfaces. There is no support for data plane Ethernet-like interfaces.

Note Set Operation on MAU-MIB objects is not supported.

Table 1-5 MAU-MIB Constraints

MIB Object	Notes
rpMauTable	Not supported
rpJackTable	Not supported
Traps (ifMauJabberTrap)	Not supported. Not applicable to CE interfaces

Cisco XR 12000 Series Router MIB Specifications

This chapter describes the MIB on the Cisco XR 12000 Series Router (C12000). Each MIB description lists any constraints on how the MIB or its object identifiers (OIDs) are implemented on the Cisco XR 12000 Series Router.

Unless noted otherwise, the Cisco XR 12000 Series Router implementation of a MIB follows the standard MIB that has been defined. Any MIB table or object not listed in the table is implemented as defined in the standard MIB definition.

This chapter contains the following sections:

- ·Cisco XR 12000 Series Router MIBs
- ·Cisco XR 12000 Series Router MIB Categories
- ·MIB Version String Description
- ·MIBs in the Cisco XR 12000 Series Router

Cisco XR 12000 Series Router MIBs

Each MIB description lists relevant constraints about the implementation of the MIB on the Cisco XR 12000 Series Router platform. Any objects not listed in a table are implemented as defined in the MIB. For detailed MIB descriptions, see the standard MIB.

Note Not all MIBs included in a Cisco IOS XR Software release are fully supported by the router. Some MIBs are not supported at all. Other MIBs might work, but they have not been tested on the router. In addition, some MIBs are deprecated but cannot be removed from the software. When a MIB is included in the image, this does not necessarily mean it is supported by the Cisco XR 12000 Series Router platform.

To determine which MIBs are included in other releases, see the “Downloading and Compiling MIBs” section.

Cisco XR 12000 Series Router MIB Categories

The MIBs in the Cisco XR 12000 Series Router are categorized into three types:

- see the “Supported and Verified MIBs” section
- see the “Supported and Unverified MIBs” section
- see the “Unverified or Unsupported MIBs” section

Supported and Verified MIBs

The MIB exists in the image, the code is implemented, and Cisco has verified that all the supported objects work properly. These MIBs are tested for the Cisco XR 12000 Series Router.

Supported and Unverified MIBs

The MIB exists in the image, the code is implemented, but we have not verified if it is working properly. In other words, the user may get something if they query the MIB. However, the information may be correct or incorrect if the MIB has not been tested. These MIBs are not tested for the Cisco XR 12000 Series Router support.

Unverified or Unsupported MIBs

The MIB exists in the image but is either not tested or not supported. These MIBs are neither tested nor supported for the Cisco XR 12000 Series Router.

MIB Version String Description

The MIB version string indicates the date and time that the module was most recently modified. The format is YYMMDDHHMMZ or YYYYMMDDHHMMZ, where:

- YY is the last two digits of the year (only years between 1900 and 1999).
- YYYY is all four digits of the year (any year).
- MM is the month (01 through 12).
- DD is the day of the month (01 through 31).
- HH is hours (00 through 23).
- MM is minutes (00 through 59).
- Z (the ASCII character Z) denotes Coordinated Universal Time (UTC, formerly Greenwich Mean Time, GMT). This datatype stores the date and time fields YEAR, MONTH, DAY, HOUR, MINUTE, SECOND, TIMEZONE_HOUR, and TIMEZONE_MINUTE.

Note For example, 9502192015Z and 199502192015Z represent 8:15 GMT on 19 February 1995. Years after 1999 use the four-digit format. Years 1900–1999 may use the two or four digit format.

Note In the following table the term Revision not available refers to the MIB module that does not have a recorded time stamp indicating the latest modification.

MIBs in the Cisco XR 12000 Series Router

This table lists the MIBs in the Cisco XR 12000 Series Router:

Table 1-1 MIBs in the Cisco XR 12000 Series Router

MIB	midb process	Supported		Unsup ported or Unverified	Not in Image
		Verified	Unverified		
CISCO-FABRIC-C12K-MIB	mibd-entity				
- Release 3.7		200209200000Z			
- Release 3.9		200209200000Z			
- Release 4.0		200209200000Z			
- Release 4.2		200209200000Z			
- Release 4.3		200209200000Z			

MIB Notification Names in the Cisco XR 12000 Series Router

This table lists the Notification Names associated with MIBs in the Cisco XR 12000 Series Router:

Table 1-2 MIB Notification Names in the Cisco XR 12000 Series Router

MIB	Notification Name
CISCO-FABRIC-C12K-MIB	ciscoFabric12kMIBFabMasterSchCh

CISCO-FABRIC-C12K-MIB

The CISCO-FABRIC-C12K-MIB is the MIB module for the c12000 series routers. This MIB module is used for managing/tracking the c12000 fabric entities and fabric related configuration, status, and statistics information.

C12000 fabric architecture is based on NxN non-blocking crossbar switch fabric, where N stands for the maximum number of LCs that can be supported in the chassis (this includes the RP). The connections through the switch fabric is controlled by a scheduler (CSC), the CSC accepts transmission requests from line cards, issues grants to access the fabric, and provides a reference clock to all the cards in the system to synchronize data transfer across the crossbar.

Some of the error status objects in the MIB are read-clear, that is the value of the object is cleared on a query of the object. Details about the various fabric related attributes specified in the MIB could be found at http://www.cisco.com/c/en/us/support/docs/routers/12000-series-routers/22281-hwts-12000-22281.html

This table lists the tables associated with this MIB.

Table 1-3 CISCO-FABRIC-C12K-MIB Tables and Descriptions

Name	**Description**
cfcGenericFabToFabTable	Table providing ToFabFIA statistics and information in the managed system.
cfcGenericFabFrFabTable	Table providing FrFabFIA information in the managed system.
cfcGenericFabFrFabSliTable	Table providing per serial link information maintained by the FrFabFIA in the managed system.
cfcGenericScaTable	Table providing SCA statistics and information in the managed system.
cfcGenericXbarTable	Table providing Xbar information in the managed system.
cfcPreOc192FabToFabTable	Table providing pre-OC192 ToFabFIA statistics and information in the managed system.
cfcPreOc192FabFrFabTable	Table providing pre-OC192 FrFabFIA statistics and information in the managed system.
cfcPreOc192ScaTable	Table providing pre-OC192 SCA statistics and information in the managed system.
cfcPreOc192XbarTable	Table providing pre-OC192 Xbar statistics and information in the managed system.
cfcOc192FabToFabTable	Entry providing various statistics and information of OC192 ToFabFIA on an associated linecard identified by entPhysicalIndex.
cfcOc192FabFrFabTable	Table providing FrFabFIA statistics and information in the managed system.
cfcOc192FabFrFabSliTable	Table providing per serial link information maintained by the OC192 FrFabFIA in the managed system.
cfcOc192FabFrFabStatTable	Table providing per module statistics information maintained by the OC192 FrFabFIA in the managed system.
cfcOc192ScaTable	Table providing OC192 SCA statistics and information in the managed system.
cfcOc192XbarTable	Table providing OC192 Xbar statistics and information in the managed system.

Monitoring Notifications

This chapter describes the Cisco ASR 9000 Series routernotifications supported by the MIB enhancements feature introduced in Cisco IOS XR Software Release 3.7. SNMP (Simple Network Management Protocol) uses notifications to report events on a managed device. The notifications are traps for different events. The router also supports other notifications not listed.

This chapter contains the following sections:

- ·SNMP Notification Overview
- ·Enabling Notifications
- ·Cisco SNMP Notifications

SNMP Notification Overview

An SNMP agent can notify the SNMP manager when important system events occur, such as the following:

- Interface or card starts or stops running
- Temperature thresholds are crossed
- Authentication failures occur

When an agent detects an alarm condition, the agent:

- Logs information about the time, type, and severity of the condition
- Generates a notification message, which it then sends to a designated IP host SNMP notifications are sent as one of the following:
  - Traps–Unreliable messages, which do not require receipt acknowledgement from the SNMP manager.

To use SNMP notifications on your system, you must specify their recipients. These recipients indicate where Network Registrar notifications are directed. By default, all notifications are enabled, but no recipients are defined. Until you define the recipients, no notifications are sent.

Many commands use the **traps** keyword in the command syntax.

Note Most notification types are disabled by default. However, some notification types cannot be controlled with the **snmp** command. For example, some notification types can be enabled by **snmp** or CLI (command-line interface) and other types are enabled by a combination of CLI and **snmp**. The linkUpDown notifications are controlled by the **snmp trap link-status** and **snmp-server trap link ietf** commands.

Specify the trap types to avoid having all traps sent, then use multiple **snmp-server traps** commands, one for each of the trap types used in the **snmp host** command.

Enabling Notifications

You can enable MIB notifications using either of the following procedures:

- CLI–Specify the recipient of the trap message and specify the types of traps sent. The enabling command also specifies which types of traps are enabled. For detailed procedures, go to the following URL: http://www.cisco.com/en/US/docs/ios_xr_sw/iosxr_r3.7/vfw/command/reference/vfr37snp.html

- Performing an SNMP SET operation with the **setany** command–To enable or disable MIB notifications, perform an SNMP SET operation on a specific object.
  - To enable the notifications, set the object to true (1).
  - To disable the notifications, set the object to false (2).

For detailed procedures, go to the following URL:
http://www.cisco.com/en/US/docs/ios_xr_sw/iosxr_r3.7/system_management/command/reference/yr37snmp.html

Note If you issue the **snmp-server traps** command without a notification-type argument, the router generates traps for all types of events, which might not be desirable. Some MIBs require the user to set additional objects to enable some notifications.

Cisco SNMP Notifications

This section contains tables that describe a MIB event, why the event occurred, and a recommendation as to how to handle the event. Each table lists the following information:

- Event–Event display
- Description–What the event indicates
- Probable cause–What might have caused the notification
- Recommended action–Recommendation as to what should be done when the particular notification occurs

Note In the following tables, where "no action is required" is documented, there might be instances where an application, such as trouble ticketing, occurs. For detailed information, go to:http://www.cisco.com/en/US/docs/ios_xr_sw/iosxr_r3.7/system_management/command/reference/yr37snmp.html

Environmental or Functional Notifications

This table lists notifications generated for events that might indicate the failure of the Cisco Carrier Routing System or conditions that might affect router functionality.

Table 1-1 Environmental or Functional Notifications

Event	Description	Probable Cause	Recommended Action
cefcModuleStatusChange	Indicates that the status of a module has changed.	Module has unknown state.	Enter the **show platform** command to view error message details. For syslog messages associated with this event, consult Messages and Recovery Procedures.
		Module is operational.	No action is required.
		Module has failed because of some condition.	Enter the show module command to view error message details. For syslog messages associated with this event, consult Messages and Recovery Procedures.
cefcPowerStatusCh	Indicates that the power	FRU is powered off	Enter the **show environment**

Table 1-1 *Environmental or Functional Notifications*

Event	Description	Probable Cause	Recommended Action
ange	status of a FRU has changed.	because of unknown problem.	command to check the actual power usage. For syslog messages associated with this event, consult Messages and Recovery Procedures.
		FRU is powered on.	No action is required.
		FRU is administratively off.	No action is required.
		FRU is powered off as the available system power is insufficient.	Enter the **show environment** command to check the actual power usage.
cefcFRUInserted	Indicates that a FRU was inserted	A new field-replaceable unit such as a fan, transceiver, power supply, or redundant power supply was added.	No action is required.
cefcFRURemoved	Indicates that a FRU was removed.	A field-replaceable unit such as a fan, transceiver, power supply, or redundant power supply was removed.	Replace the field-replaceable unit.
cPimNbrLoss	Indicates the loss of an adjacency with a neighbor.	The neighbor timer expires, and the router has no other neighbors on the same interface with a lower IP address than itself.	Use the CLI command **show tech-support multicast address-family {ipv4 | ipv6} vrf vrf-**name to gather information for the affected VRF when the traps are received.

Flash Card Notifications

This table lists CISCO-FLASH-MIB notifications generated by Cisco Carrier Routing System flash cards. These notifications indicate the failure of a flash card or error conditions on the card that might affect the functionality of all interfaces.

Table 1-2 *Flash Card Notifications*

Event	Description	Probable Cause	Recommended Action
ciscoFlashDeviceInsertedNotif	Indicates a removable flash device was inserted into the router.	A removable flash device was inserted into the router.	Check for the flash card that was inserted from ciscoFlashDeviceTable. This information is also provided in the notification itself.
ciscoFlashDeviceRemovedNotif	Indicates a removable flash card was removed from the	A removable flash device was removed from the router.	Check the ciscoFlashDeviceTable to identify the removed flash card. This information is also

Table 1-2 Flash Card Notifications

Event	Description	Probable Cause	Recommended Action
	router.		provided in the notification itself.

Interface Notifications

This table lists notifications generated by the router for link-related (interface) events.

Table 1-3 Interface Notifications

Event	Description	Probable Cause	Recommended Action
linkDown	Indicates that a link is about to enter the down state, which means it cannot transmit or receive traffic. The ifOperStatus object shows the previous state of the link. Value is down(2).	An internal software error might have occurred.	Use the CLI command, **show interface MgmtEth 0/2/CPU0/0**, to determine the cause of the interface down.
linkUp	Indicates that the link is up. The value of ifOperStatus indicates the new link state. Value is up(1).	The port manager reactivated a port in the linkdown state during a switchover.	No action is required.

Routing Protocol Notifications

This table lists BGP4-MIB notifications that are Border Gateway Protocol (BGP) state changes generated by the Cisco Carrier Routing System to indicate error conditions for routing protocols and services.

Table 1-4 Routing Protocol Notifications

Event	Description	Probable Cause	Recommended Action
bgpEstablished	The BGP FSM[1] enters the ESTABLISHED state. It becomes active on the router.	The BGP routing protocol changed status.	No action is required.
bgpBackwardTransition	Indicates BGP protocol transition from a higher-level state to a lower-level state. The prefix count for an address family on a BGP session exceeded the configured threshold value.	The BGP routing protocol changed status.	This threshold value is configured using the CLI command **neighbor nbr_addr max_prefixes** [*threshold*] [*warning-only*]

[1] FSM = finite state machine

Redundancy Framework Notifications

This table lists CISCO-RF-MIB notifications that can occur in a redundant system. There are two types of notifications:

- Switch of Activity (SWACT)–Either a forced or automatic switch of active status from the active unit to the standby unit. The former standby unit is now referred to as the active unit.
- Progression–Process of making the redundancy state of the standby unit equivalent to that of the active unit. This includes transitioning the RF state machine through several states, which drives the clients on the active unit to synchronize any relevant data with their peer on the standby unit.

Table 1-5 Redundancy Framework Notifications

Event	Description	Probable Cause	Recommended Action
ciscoRFSwactNotif	Indicates that the RF state changed. A switch of activity notification is sent by the newly active redundant unit.	A switch of activity occurs. If a SWACT event is indistinguishable from a reset event, a network management station should use this notification to differentiate the activity.	If the switchover occurred because the active unit failed (indicated by cRFStatusLastSwactReasonCode), see if there are any hardware failures; otherwise, no action is required.
ciscoRFProgressionNotif	Indicates that the RF state changed.	The active redundant unit RF state changed or the RF state of the peer unit changed.	To avoid an increase of notifications for all state transitions, send notifications for transitions to the following RF states: - standbyCold(5) - standbyHot(9) - active(14) - activeExtraload(15)

Using MIBs

This chapter describes how to work with MIBs on the Cisco Carrier Routing System. This appendix contains the following sections:

- Cisco Unique Device Identifier Support
- Cisco Redundancy Features
- Managing Physical Entities
- Monitoring Quality of Service
- Monitoring Router Interfaces
- Billing Customers for Traffic
- Using IF-MIB Counters

Cisco Unique Device Identifier Support

The ENTITY-MIB supports the Cisco compliance effort for a unique device identifier (UDI) standard stored in Identification Programmable Read-Only Memory (IDPROM).

The Cisco UDI provides a unique identity for every Cisco product. The UDI is composed of three separate data elements that must be stored in the entPhysicalTable:

- Orderable product identifier (PID)–Alphanumeric identifier used by customers to order Cisco products. Two examples include A9K-RSP-4G and A9K-4T-E. PID is limited to 18 characters and must be stored in the entPhysicalModelName object.
- Version identifier (VID)–Version of the PID. The VID indicates the number of times a product has versioned in ways that are reported to a customer. For example, the product identifier A9K-RSP-4G may have a VID of V04. VID is limited to three alphanumeric characters and must be stored in the entPhysicalHardwareRev object.
- Serial number (SN)–11-character identifier used to identify a specific part within a product and must be stored in the entPhysicalSerialNum object. Serial number content is defined by manufacturing part number 7018060-0000. The SN is accessed at the following website by searching on the part number 701806-0000:

  https://tools.cisco.com/emco/inbiz/inbiz/Home

  Serial number format is defined in four fields:

  — Location (L)

  — Year (Y)

  — Workweek (W)

  — Sequential serial ID (S)

  The SN label is represented as: LLLYYWWSSS.

Note The Version ID returns NULL for those old or existing cards with IDPROMs that do not have the Version ID field. Therefore, corresponding entPhysicalHardwareRev returns NULL for cards that do not have the Version ID field in IDPROM.

Cisco Redundancy Features

Redundancy creates a duplication of data elements and software functions to provide an alternative in case of failure. The goal of Cisco redundancy features is to cut over without affecting the link and protocol states associated with each interface and continue packet forwarding. The state of the interfaces and subinterfaces is maintained, along with the state of line cards and various packet processing hardware.

This section describes Cisco redundancy feature:

- Levels of Redundancy

Levels of Redundancy

This section describes the levels of redundancy supported on the Cisco Carrier Routing System and how to verify that this feature is available. The Cisco Carrier Routing System supports fault resistance by allowing a Cisco redundant Route Switch Processor (RSP) to take over if the active RSP fails. Redundancy prevents equipment failures from causing service outages, and supports hitless maintenance and upgrade activities. The state of the interfaces and subinterfaces is maintained along with the state of line cards and various packet processing hardware.

Redundant systems support two RSP. One acts as the active RSPs while the other acts as the standby RSPs.

The redundancy feature provides high availability for the Cisco routers by switching when one of the following conditions occur:

- Cisco IOS XR Software failure
- Software upgrade
- Maintenance procedure

The Cisco Carrier Routing System operates in Nonstop Forwarding/Stateful Switchover (NSF/SSO) mode.

Nonstop Forwarding/Stateful Switchover

This section describes the Nonstop Forwarding/Stateful Switchover mode. With NSF/SSO, the Cisco Carrier Routing System can change from the active to the standby RSPs almost immediately while continuing to forward packets. Cisco IOS XR Software NSF/SSO support on this platform enables immediate switchover.

In networking devices running NSF/SSO, both RSPs must be running the same configuration so that the standby RSP is always ready to assume control following a fault on the active RSP. The configuration information is synchronized from the active RSP to the standby RSP at startup and each timechanges to the active RSP configuration occur.

Following an initial synchronization between the two RSPs, NSF/SSO maintains RSP state information between them, including forwarding information.

The Cisco Nonstop Forwarding (NSF) works with Stateful Switchover (SSO) to minimize the amount of time a network is unavailable to its users following a Route Switching Processor (RSP) fail-over in a router with dual RSPs. NSF/SSO capability allows routers to detect a switchover and take the necessary actions to continue forwarding network traffic and to recover route information from peer devices.

The Cisco NSF works with the Stateful Switchover (SSO) feature in Cisco IOS XR Software to minimize the amount of time a network is unavailable to its users following a switchover. The main objective of the Cisco NSF/SSO is to continue forwarding data packets along known routes while the routing protocol information is restored following a route switchover.

Managing Physical Entities

This section describes how to use SNMP to manage the physical entities (components) in the router by:

- Performing Inventory Management
- Monitoring and Configuring FRU Status
- Generating SNMP Notifications

Purpose and Benefits

The physical entity management feature of the Cisco Carrier Routing System SNMP implementation does the following:

- Monitors and configures the status of field-replaceable units (FRUs)
- Provides information about physical port to interface mappings
- Provides asset information for asset tagging
- Provides firmware and software information for chassis components

MIBs Used for Physical Entity Management

- CISCO-ENTITY-ASSET-MIB—Contains asset tracking information (IDPROM contents) for the physical entities listed in the entPhysicalTable of the ENTITY-MIB. The MIB provides device-specific information for physical entities, including orderable part number, serial number, manufacturing assembly number, and hardware, software, and firmware information.
- CISCO-ENTITY-FRU-CONTROL-MIB—Contains objects used to monitor and configure the administrative and operational status of field-replaceable units (FRUs), such as fans, RSPs, and transceivers that are listed in the entPhysicalTable of the ENTITY-MIB.
- CISCO-ENTITY-SENSOR-MIB—Contains information about entities in the entPhysicalTable with an entPhysicalClass value of sensor.
- ENTITY-MIB—Contains information for managing physical entities on the router. It also organizes the entities into a containment tree that depicts their hierarchy and relationship to each other. The MIB contains the following tables:
  - The entPhysicalTable describes each physical component (entity) in the router. The table contains an entry for the top-level entity (the chassis) and for each entity in the chassis. Each entry provides information about that entity: its name, type, vendor, and a description, and a description of how the entity fits into the hierarchy of chassis entities.

    Each entity is identified by a unique index (entPhysicalIndex) that is used to access information about the entity in this and other MIBs.
  - The entAliasMappingTable maps each physical port's entPhysicalIndex value to its corresponding ifIndex value in the IF-MIB ifTable.
  - The entPhysicalContainsTable shows the relationship between physical entities in the chassis. For each physical entity, the table lists the entPhysicalIndex for each of the entity's child objects.

Performing Inventory Management

To get information about entities in the router, perform a MIB walk on the ENTITY-MIB entPhysicalTable. As you examine sample entries in the ENTITY-MIB entPhysicalTable, consider the following objects:

- entPhysicalIndex–Uniquely identifies each entity in the chassis. This index is also used to access information about the entity in other MIBs.
- entPhysicalContainedIn–Indicates the entPhysicalIndex of a component parent entity.
- entPhysicalParentRelPos–Shows the relative position of same-type entities that have the same entPhysicalContainedIn value (for example, chassis slots, and line card ports).

Note The container is applicable if the physical entity class is capable of containing one or more removable physical entities. For example, each (empty or full) slot in a chassis is modeled as a container. All removable physical entities should be modeled within a container entity, such as field-replaceable modules, fans, or power supplies.

Sample of ENTITY-MIB entPhysicalTable Entries

The samples in this section show how information is stored in the entPhysicalTable. You can perform asset inventory by examining entPhysicalTable entries.

Note The sample outputs and values that appear throughout this appendix are examples of data you can view when using MIBs.

The following display shows the ENTITY-MIB entPhysicalTable sample entries:

```
entPhysicalDescr.186 = 4-Port 10GE        Extended Line Card, Requires XFPs
entPhysicalDescr.187 = Ten GigabitEthernet Port
entPhysicalDescr.188 = GigeEthernet XFP container
entPhysicalDescr.189 =
entPhysicalDescr.190 = Transceiver        Temperature Sensor
entPhysicalDescr.191 = Transceiver        Tx PowerSensor
entPhysicalDescr.192 = Transceiver        Rx PowerSensor
entPhysicalDescr.193 = Transceiver        Transmit Bias Current Sensor
entPhysicalDescr.194 = Line Card host
entPhysicalDescr.195 = Inlet Temperature Sensor entPhysicalDescr.196 = Hot Temperature Sensor entPhysicalDescr.197 = Voltage
Sensor - IBV entPhysicalDescr.198 = Voltage Sensor - 5.0V entPhysicalDescr.199 = Voltage Sensor - VP3P3_CAN
entPhysicalDescr.200 = Voltage Sensor - 3.3V
```

Where **entPhysicalDescr** identifies the manufacturer name for the physical entity.

```
entPhysicalVendorType.186 = cevModuleA9K4x10GEE entPhysicalVendorType.187 =
cevPortGEXFP entPhysicalVendorType.188 = cevContainerXFP entPhysicalVendorType.189 =
cevXFPUnknown entPhysicalVendorType.190 = cevSensorTransceiverTemp
entPhysicalVendorType.191 = cevSensorTransceiverTxPwr entPhysicalVendorType.192 =
cevSensorTransceiverRxPwr entPhysicalVendorType.193 = cevSensorTransceiverCurrent
entPhysicalVendorType.194 = cevModuleASR9KHost entPhysicalVendorType.195 =
cevSensorModuleInletTemp entPhysicalVendorType.196 = cevSensorHotTemperature
entPhysicalVendorType.197 = cevSensorModuleDeviceVoltage entPhysicalVendorType.198 =
cevSensorModuleDeviceVoltage entPhysicalVendorType.199 = cevSensorModuleDeviceVoltage
entPhysicalVendorType.200 = cevSensorModuleDeviceVoltage
```

Where **entPhysicalVendorType** identifies the unique vendor-specific hardware type of the physical entity.

```
entPhysicalContainedIn.186 = 92 entPhysicalContainedIn.187 = 186
entPhysicalContainedIn.188 = 187 entPhysicalContainedIn.189 =
188 entPhysicalContainedIn.190 = 189 entPhysicalContainedIn.191
= 189 entPhysicalContainedIn.192 = 189
entPhysicalContainedIn.193 = 189 entPhysicalContainedIn.194 =
186 entPhysicalContainedIn.195 = 194 entPhysicalContainedIn.196
= 194 entPhysicalContainedIn.197 = 194
entPhysicalContainedIn.198 = 194 entPhysicalContainedIn.199 =
194 entPhysicalContainedIn.200 = 194
```

Where **entPhysicalContainedIn** indicates the entPhysicalIndex of a parent entity (component).

```
entPhysicalClass.186 = module(9) entPhysicalClass.187 = port(10)
entPhysicalClass.188 = container(5) entPhysicalClass.189 = module(9)
entPhysicalClass.190 = sensor(8) entPhysicalClass.191 = sensor(8)
entPhysicalClass.192 = sensor(8) entPhysicalClass.193 = sensor(8)
entPhysicalClass.194 = module(9) entPhysicalClass.195 = sensor(8)
entPhysicalClass.196 = sensor(8) entPhysicalClass.197 = sensor(8)
entPhysicalClass.198 = sensor(8) entPhysicalClass.199 = sensor(8)
entPhysicalClass.200 = sensor(8)
```

Where **entPhysicalClass** indicates the general type of hardware device.

```
entPhysicalParentRelPos.186 = 0 entPhysicalParentRelPos.187 =
1 entPhysicalParentRelPos.188 = 0 entPhysicalParentRelPos.189
= 0 entPhysicalParentRelPos.190 = 0
entPhysicalParentRelPos.191 = 1 entPhysicalParentRelPos.192 =
2 entPhysicalParentRelPos.193 = 3 entPhysicalParentRelPos.194
= 0 entPhysicalParentRelPos.195 = 0
entPhysicalParentRelPos.196 = 1 entPhysicalParentRelPos.197 =
2 entPhysicalParentRelPos.198 = 3 entPhysicalParentRelPos.199
= 4 entPhysicalParentRelPos.200 = 5
```

Where **entPhysicalParentRelPos** indicates the relative position of this child among the other entities.

```
entPhysicalName.186 = module 0/5/CPU0 entPhysicalName.187 =
TenGigE0/5/0/1 entPhysicalName.188 = slot mau 0/5/CPU0/1
entPhysicalName.189 = module mau 0/5/CPU0/1 entPhysicalName.190 =
temperature 0/5/CPU0/1 entPhysicalName.191 = power Tx 0/5/CPU0/1
entPhysicalName.192 = power Rx 0/5/CPU0/1 entPhysicalName.193 = current
0/5/CPU0/1 entPhysicalName.194 = module 0/5/CPU0 entPhysicalName.195 =
temperature 0/5/CPU0 entPhysicalName.196 = temperature 0/5/CPU0
entPhysicalName.197 = voltage 0/5/CPU0 entPhysicalName.198 = voltage
0/5/CPU0 entPhysicalName.199 = voltage 0/5/CPU0 entPhysicalName.200 =
voltage 0/5/CPU0
```

Where **entPhysicalName** provides the textual name of the physical entity.

```
entPhysicalHardwareRev.186 = entPhysicalHardwareRev.187
= entPhysicalHardwareRev.188 =
entPhysicalHardwareRev.189 = entPhysicalHardwareRev.190
= entPhysicalHardwareRev.191 =
entPhysicalHardwareRev.192 = entPhysicalHardwareRev.193
= entPhysicalHardwareRev.194 =
entPhysicalHardwareRev.195 = entPhysicalHardwareRev.196
= entPhysicalHardwareRev.197 =
entPhysicalHardwareRev.198 = entPhysicalHardwareRev.199
= entPhysicalHardwareRev.200 =
```

Where **entPhysicalHardwareRev** provides the vendor-specific hardware revision number (string) for the physical entity.

```
entPhysicalFirmwareRev.186 = Version 0.63(20081010:215422)
entPhysicalFirmwareRev.187 =
entPhysicalFirmwareRev.188 =
entPhysicalFirmwareRev.189   =   entPhysicalFirmwareRev.190   =
entPhysicalFirmwareRev.191   =   entPhysicalFirmwareRev.192   =
entPhysicalFirmwareRev.193   =   entPhysicalFirmwareRev.194   =
entPhysicalFirmwareRev.195   =   entPhysicalFirmwareRev.196   =
entPhysicalFirmwareRev.197   =   entPhysicalFirmwareRev.198   =
entPhysicalFirmwareRev.199 = entPhysicalFirmwareRev.200 =
```

Where **entPhysicalFirmwareRev** provides the vendor-specific firmware revision number (string) for the physical entity.

```
entPhysicalSoftwareRev.186 = 3.7.2.24I entPhysicalSoftwareRev.187 =
entPhysicalSoftwareRev.188 = entPhysicalSoftwareRev.189 = 3.7.2.24I
entPhysicalSoftwareRev.190 = entPhysicalSoftwareRev.191 =
entPhysicalSoftwareRev.192 = entPhysicalSoftwareRev.193 =
entPhysicalSoftwareRev.194 = 3.7.2.24I entPhysicalSoftwareRev.195 =
entPhysicalSoftwareRev.196 = entPhysicalSoftwareRev.197 =
entPhysicalSoftwareRev.198 = entPhysicalSoftwareRev.199 =
entPhysicalSoftwareRev.200 =
```

Where **entPhysicalSoftwareRev** provides the software revision number for the physical entity.

```
entPhysicalSerialNum.186 = FHH1213002A entPhysicalSerialNum.187 =
entPhysicalSerialNum.188 = entPhysicalSerialNum.189 = ECL114704JD
entPhysicalSerialNum.190 = entPhysicalSerialNum.191 =
entPhysicalSerialNum.192 = entPhysicalSerialNum.193 =
entPhysicalSerialNum.194 = entPhysicalSerialNum.195 =
entPhysicalSerialNum.196 = entPhysicalSerialNum.197 =
entPhysicalSerialNum.198 = entPhysicalSerialNum.199 =
entPhysicalSerialNum.200 =
```

Where **entPhysicalSerialNum** provides the vendor-specific serial number (string) for the physical entity.

```
entPhysicalMfgName.186 = Cisco Systems Inc. entPhysicalMfgName.187 =
entPhysicalMfgName.188 = entPhysicalMfgName.189 =
entPhysicalMfgName.190 = entPhysicalMfgName.191 =
entPhysicalMfgName.192 = entPhysicalMfgName.193 =
entPhysicalMfgName.194   =   entPhysicalMfgName.195   =
entPhysicalMfgName.196   =   entPhysicalMfgName.197   =
entPhysicalMfgName.198   =   entPhysicalMfgName.199   =
entPhysicalMfgName.200 =
```

Where **entPhysicalMfgName** provides the manufacturer name for the physical component.

```
entPhysicalModelName.186 = A9K-4T-E entPhysicalModelName.187 =
entPhysicalModelName.188 = entPhysicalModelName.189 = ONS-XC-10G-
S1 entPhysicalModelName.190 = entPhysicalModelName.191 =
entPhysicalModelName.192 = entPhysicalModelName.193 =
entPhysicalModelName.194 = entPhysicalModelName.195 =
entPhysicalModelName.196 = entPhysicalModelName.197 =
entPhysicalModelName.198 = entPhysicalModelName.199 =
entPhysicalModelName.200 =
```

Where **entPhysicalModelName** provides the vendor-specific model name string for the physical component.

```
entPhysicalAlias.186 = entPhysicalAlias.187 =
entPhysicalAlias.188 = entPhysicalAlias.189 =
entPhysicalAlias.190 = entPhysicalAlias.191 =
entPhysicalAlias.192 = entPhysicalAlias.193 =
entPhysicalAlias.194 = host entPhysicalAlias.195 =
entPhysicalAlias.196 = entPhysicalAlias.197 =
entPhysicalAlias.198 = entPhysicalAlias.199 =
entPhysicalAlias.200 =
```

Where **entPhysicalAlias** provides the alias name for the physical component.

```
entPhysicalAssetID.186     =     entPhysicalAssetID.187     =
entPhysicalAssetID.188     =     entPhysicalAssetID.189     =
entPhysicalAssetID.190     =     entPhysicalAssetID.191     =
entPhysicalAssetID.192     =     entPhysicalAssetID.193     =
entPhysicalAssetID.194     =     entPhysicalAssetID.195     =
entPhysicalAssetID.196     =     entPhysicalAssetID.197     =
entPhysicalAssetID.198     =     entPhysicalAssetID.199     =
entPhysicalAssetID.200 =
```

Where **entPhysicalAssetID** provides the vendor-specific asset ID for the physical component.

```
entPhysicalIsFRU.186 = true(1) entPhysicalIsFRU.187 = false(2)
entPhysicalIsFRU.188 = false(2) entPhysicalIsFRU.189 = true(1)
entPhysicalIsFRU.190 = false(2) entPhysicalIsFRU.191 = false(2)
entPhysicalIsFRU.192 = false(2) entPhysicalIsFRU.193 = false(2)
entPhysicalIsFRU.194 = false(2) entPhysicalIsFRU.195 = false(2)
entPhysicalIsFRU.196 = false(2) entPhysicalIsFRU.197 = false(2)
entPhysicalIsFRU.198 = false(2) entPhysicalIsFRU.199 = false(2)
entPhysicalIsFRU.200 = false(2)
```

Where **entPhysicalIsFRU** indicates whether or not this physical entity is considered a field-replaceable unit (FRU).

Note the following about the sample configuration:

- All chassis slots and line card ports have the same entPhysicalContainedIn value:
  - For chassis slots, entPhysicalContainedIn = 1 (the entPhysicalIndex of the chassis).
  - For line card ports, entPhysicalContainedIn = 26 (the entPhysicalIndex of the line card).
- Each chassis slot and line card port has a different entPhysicalParentRelPos to show its relative position within the parent object.

Determining the ifIndex Value for a Physical Port

The ENTITY-MIB entAliasMappingIdentifier maps a physical port to an interface by mapping the port's entPhysicalIndex to its corresponding ifIndex value in the IF-MIB ifTable. The following sample shows that the physical port with a entPhysicalIndex value of 35 is associated with the interface with the ifIndex value of four:

entAliasMappingIdentifer.**35**.0 = ifIndex.**4**

Note See the MIB for detailed descriptions of possible MIB values.

Monitoring and Configuring FRU Status

View objects in the CISCO-ENTITY-FRU-CONTROL-MIB cefcModuleTable to determine the administrative and operational status of FRUs, such as power supplies and line cards:

- cefcModuleAdminStatus–administrative state of the FRU. This object is read-only and returns enable.
- cefcModuleOperStatus–current operational state of the FRU.

This figure shows a cefcModuleTable entry for a line card with the entPhysicalIndex value of 24.

Figure 1-1 Sample cefcModuleTable Entry

```
cefcModuleEntry.entPhysicalIndex
cefcModuleEntry.24
  cefcModuleAdminStatus = enabled(1)
  cefcModuleOperStatus = ok(2)
  cefcModuleResetReason = manual
reset(5)
  cefcModuleStatusLastChangeTime =
7714
```

See the "FRU Status Changes" section for information about how the router generates notifications to indicate changes in FRU status.

Generating SNMP Notifications

This section provides information about the SNMP notifications generated in response to events and conditions on the router, and describes how to identify which hosts are to receive notifications.

- Identifying Hosts to Receive Notifications
- Configuration Changes
- FRU Status Changes

Identifying Hosts to Receive Notifications

You can use the CLI or SNMP to identify hosts to receive SNMP notifications and to specify the types of notifications they are to receive (notifications). For CLI instructions, see the “Enabling Notifications” section. To use SNMP to configure this information:

Use SNMP-NOTIFICATION-MIB objects, including the following examples, to select target hosts and specify the types of notifications to generate for those hosts:

- snmpNotifyTable–Contains objects to select hosts and notification types:
  - snmpNotifyTag is an arbitrary octet string (a tag value) used to identify the hosts to receive SNMP notifications. Information about target hosts is defined in the snmpTargetAddrTable (SNMP-TARGET-MIB), and each host has one or more tag values associated with it. If a host in snmpTargetAddrTable has a tag value that matches this snmpNotifyTag value, the host is selected to receive the types of notifications specified by snmpNotifyType.
  - snmpNotifyType is the type of SNMP notification to send: notification(1) or inform(2).
- snmpNotifyFilterProfileTable and snmpNotifyFilterTable–Use objects in these tables to create notification filters to limit the types of notifications sent to target hosts.

Use SNMP-TARGET-MIB objects to configure information about the hosts to receive notifications:

- snmpTargetAddrTable–Transport addresses of hosts to receive SNMP notifications. Each entry provides information about a host address, including a list of tag values:
  - - snmpTargetAddrTagList–set of tag values associated with the host address. If a host tag value matches snmpNotifyTag, the host is selected to receive the types of notifications defined by snmpNotifyType.
- snmpTargetParamsTable–SNMP parameters to use when generating SNMP notifications.

Use the notification enable objects in appropriate MIBs to enable and disable specific SNMP notifications.

Configuration Changes

If entity notifications are enabled, the router generates an entConfigChange notification (ENTITY-MIB) when the information in any of the following tables changes (which indicates a change to the router configuration):

- entPhysicalTable
- entAliasMappingTable
- entPhysicalContainsTable

Note A management application that tracks configuration changes checks the value of the entLastChangeTime object to detect any entConfigChange notifications that were missed as a because of throttling or transmission loss.

Enabling Notifications for Configuration Changes

To configure the router to generate an entConfigChange notification each time its configuration changes, enter the **snmp-sever trap entity** command from the CLI. Use the **no** form of the command to disable the notifications.

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

FRU Status Changes

If FRU notifications are enabled, the router generates the following notifications in response to changes in the status of a FRU:

- cefcModuleStatusChange–The operational status (cefcModuleOperStatus) of a FRU changes.
- cefcFRUInserted–A FRU is inserted in the chassis. The notification indicates the entPhysicalIndex of the FRU and the container in which it was inserted.
- cefcFRURemoved–A FRU is removed from the chassis. The notification indicates the entPhysicalIndex of the FRU and the container from which it was removed.

Note See the CISCO-ENTITY-FRU-CONTROL-MIB for more information about these notifications

Enabling FRU Notifications

To configure the router to generate notifications for FRU events, enter the **snmp-server traps fru-ctrl** command from the CLI. Use the **no** form of the command to disable the notifications.

```
Router(config)# snmp-server traps fru-ctrl
Router(config)# no snmp-server traps fru-ctrl
```

To enable FRU notifications through SNMP, set cefcMIBEnableStatusNotification to true (1). Disable the notifications by setting cefcMIBEnableStatusNotification to false (2).

Monitoring Quality of Service

This section provides the following information about using Quality of Service (QoS) in configuration:

- Cisco Carrier Routing System QoS Basics
- CISCO-CLASS-BASED-QOS-MIB Overview
- Viewing QoS Configuration Settings Using the CISCO-CLASS-BASED-QOS-MIB
- Monitoring QoS Using the CISCO-CLASS-BASED-QOS-MIB
- Considerations for Processing QoS Statistics
- Sample QoS Applications

Cisco Carrier Routing System QoS Basics

The Cisco Carrier Routing System distributes QoS features across the line cards. Line cards are designed to provide QoS features on packets that flow through the line cards.

CISCO-CLASS-BASED-QOS-MIB Overview

The CISCO-CLASS-BASED-QOS-MIB provides read-only access to QoS configuration information and statistics for Cisco platforms that support the modular Quality of Service command-line interface.

CISCO-CLASS-BASED-QOS-MIB Object Relationship

To understand how to navigate the CISCO-CLASS-BASED-QOS-MIB tables, it is important to understand the relationship among different QoS objects. QoS objects consists of:

- Match statement–Specific match criteria to identify packets for classification purposes.
- Class map–User-defined traffic class that contains one or more match statements used to classify packets into different categories.
- Feature action–Action taken on classified traffic. Features include police, traffic shaping, queueing, random detect, and packet marking. After the traffic is classified, actions are applied to packets matching each traffic class.
- Policy map–User-defined policy that associates QoS feature actions to user-defined class maps as policy maps can have multiple class maps.
- Service policy–Policy map that has been attached to an interface.

The MIB uses the following indices to identify QoS features and distinguish among instances of those features:

- cbQosObjectsIndex–Identifies each QoS feature on the router.
- cbQoSConfigIndex–Identifies a type of QoS configuration. This index is shared by QoS objects that have identical configurations.
- cbQosPolicyIndex–Identifies a unique service policy.

QoS MIB Information Storage

CISCO-CLASS-BASED-QOS-MIB information is stored as:

- Configuration information–Includes all the QoS configuration objects, such as class maps, policy map, match statements, and feature action configuration parameters. The configuration may have multiple identical instances. Configuration objects are identified by cbQosConfigIndex attribute. Multiple instances of the same QoS feature share a single configuration object that is identified by the same cbQosConfigIndex value.
- Service-policy information–Includes instances of all QoS objects, such as service-policies, classes, match statements, and feature actions. Service-policies are identified by cbQosPolicyIndex and instances of QoS objects are identified by the combination of cbQosPolicyIndex and cbQosObjectsIndex attributes.

QoS Hardware Configuration and Statistic Support

The CISCO-CLASS-BASED-QOS-MIB does not cover all the Cisco Carrier Routing System QoS hardware configuration and statistics.

The Cisco Carrier Routing System supports the concept of 'shared policy instance' where, based on the configuration, the resources for individual service policies are shared among multiple interfaces. The cbQosMIB attribute does not indicate whether the service-policies are shared-policy instances or non-shared policy instances.

The interfaces associated with the shared policy instance have a separate entry in the cbQosServicePolicyTable. The MIB entries, associated with each interface that is a part of the same shared-policy-instance, have the same data values, for example, everything except for the cbQosServicePolicyTable is identical for the rows associated with the values of cbQosPolicyIndex for such interfaces.

This figure shows how the indexes provide access to QoS configuration information and statistics.

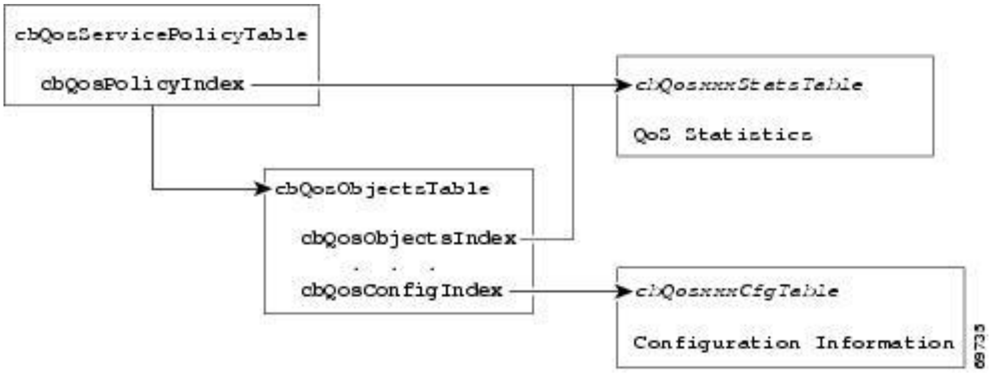


Figure 1-2 The Cisco Carrier Routing System QoS Indexes

Accessing QoS Configuration Information

To access QoS configuration information and statistics for a particular QoS feature:

Step 1 Look in cbQosServicePolicyTable and find the cbQosPolicyIndex assigned to the policy in which the feature is used.

Step 2 Use cbQosPolicyIndex to access the cbQosObjectsTable, and find the cbQosObjectsIndex and cbQosConfigIndex assigned to the QoS feature.

a.Use cbQosConfigIndex to access configuration tables (cbQosxxxCfgTable) for information about the QoS feature.

b. Use cbQosPolicyIndex and cbQosObjectsIndex to access QoS statistics tables (cbQosxxxStatsTable) for information about the QoS feature.

Viewing QoS Configuration Settings Using the CISCO-CLASS-BASED-QOS-MIB

This section contains an example that shows how QoS configuration settings are stored in CISCO-CLASS-BASED-QOS-MIB tables. The sample shows information grouped by QoS object; however, the actual output of an SNMP query might show QoS information similar to the following.

Note This is only a partial display of all QoS information.

```
ASR 9000# getmany -v3 10.86.0.94 test-user ciscoCBQosMIB CbQosServicePolicyTable cbQosIfType.1047 =
subInterface(2)
cbQosIfType.1052 = subInterface(2) cbQosPolicyDirection.1047 =
input(1) cbQosPolicyDirection.1052 = output(2) cbQosIfIndex.1047 =
36 cbQosIfIndex.1052 = 36 cbQosFrDLCI.1047 = 0
cbQosFrDLCI.1052 = 0 cbQosAtmVPI.1047 = 0 cbQosAtmVPI.1052 =
0 cbQosAtmVCI.1047 = 0 cbQosAtmVCI.1052 = 0
cbQosConfigIndex.1047.1047 = 1045 cbQosConfigIndex.1047.1048 =
1025 cbQosConfigIndex.1047.1050 = 1027
cbQosConfigIndex.1047.1051 = 1046 cbQosConfigIndex.1052.1052 =
1045 cbQosConfigIndex.1052.1053 = 1025
cbQosConfigIndex.1052.1055 = 1027 cbQosConfigIndex.1052.1056 =
1046
cbQosObjectsType.1047.1047 = policymap(1) cbQosObjectsType.1047.1048 =
classmap(2) cbQosObjectsType.1047.1050 = matchStatement(3)
cbQosObjectsType.1047.1051 = police(7) cbQosObjectsType.1052.1052 =
policymap(1) cbQosObjectsType.1052.1053 = classmap(2)
cbQosObjectsType.1052.1055 = matchStatement(3)
cbQosObjectsType.1052.1056 = police(7) cbQosParentObjectsIndex.1047.1047 =
0 cbQosParentObjectsIndex.1047.1048 = 1047
cbQosParentObjectsIndex.1047.1050 = 1048
cbQosParentObjectsIndex.1047.1051 = 1048
cbQosParentObjectsIndex.1052.1052 = 0 cbQosParentObjectsIndex.1052.1053 =
1052 cbQosParentObjectsIndex.1052.1055 = 1053
cbQosParentObjectsIndex.1052.1056 = 1053 cbQosPolicyMapName.1045 =
pm-1Meg cbQosPolicyMapDesc.1045 = cbQosCMName.1025 = class-
default cbQosCMDesc.1025=
cbQosCMInfo.1025 = matchAny(3)
. . .
```

Monitoring QoS Using the CISCO-CLASS-BASED-QOS-MIB

This section describes how to monitor QoS on the router by checking the QoS statistics in the CISCO-CLASS-BASED-QOS-MIB tables.

Note The CISCO-CLASS-BASED-QOS-MIB may contain more information than what is displayed in the output of CLI show commands.

This table lists the types of QoS statistics tables.

Table 1-1 ***QoS Statistics Tables***

QoS Table	**Statistics**
cbQosCMStatsTable	Class map–Counts of packets, bytes, and bit rate before and after QoS policies are executed. Counts of dropped packets and bytes.
cbQosPoliceStatsTable	Police action–Counts of packets, bytes, and bit rate that conforms to, exceeds, and violates police actions.
cbQosQueueingStatsTable	Queueing–Counts of discarded packets and bytes, and queue depths.
cbQosTSStatsTable	Traffic shaping–Counts of delayed and dropped packets and bytes, the state of a feature, and queue size.
cbQosREDClassStatsTable	Random early detection–Counts of packets and bytes dropped when queues are full, and counts of bytes and octets transmitted.

Considerations for Processing QoS Statistics

The router maintains 64-bit counters for most QoS statistics. However, some QoS counters are implemented as a 32-bit counter with a 1-bit overflow flag. In the following samples, the counters are shown as 33-bit counters.

When accessing QoS counter statistics, consider the following:

- SNMPv2c or SNMPv3 applications–Access the entire 64 bits of the QoS counter through cbQosxxx64 MIB objects.
- SNMPv1 applications–Access QoS statistics in the MIB as follows:
  - Access the lower 32 bits of the counter through cbQosxxx MIB objects.
  - Access the upper 32 bits of the counter through cbQosxxxOverflow MIB objects.

Sample QoS Statistics Tables

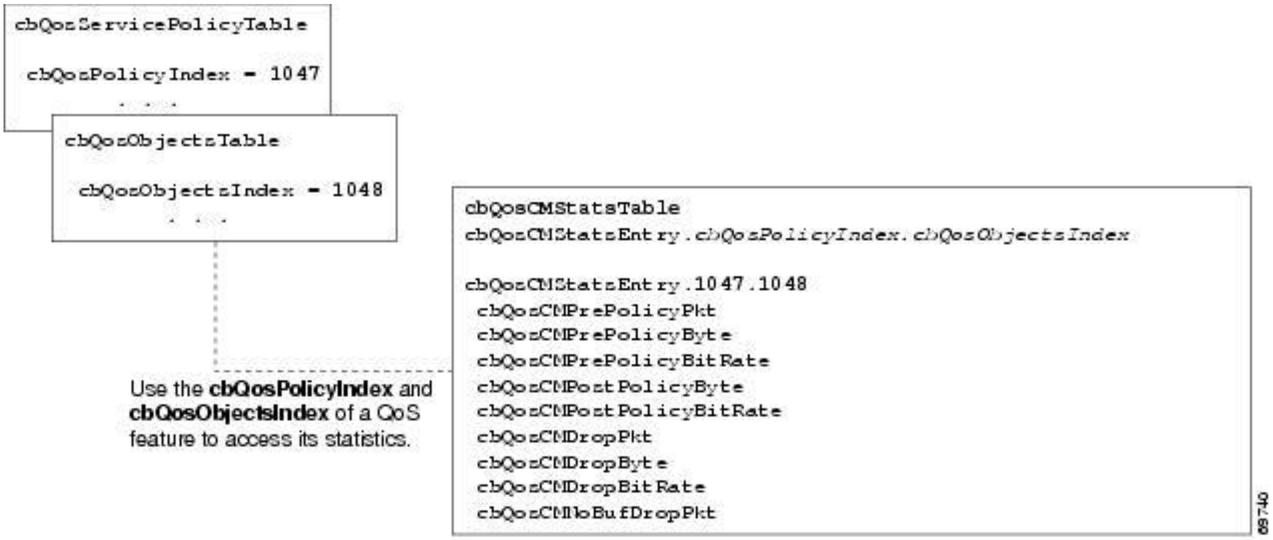
The samples in this section show the counters in CISCO-CLASS-BASED-QOS-MIB statistics tables:

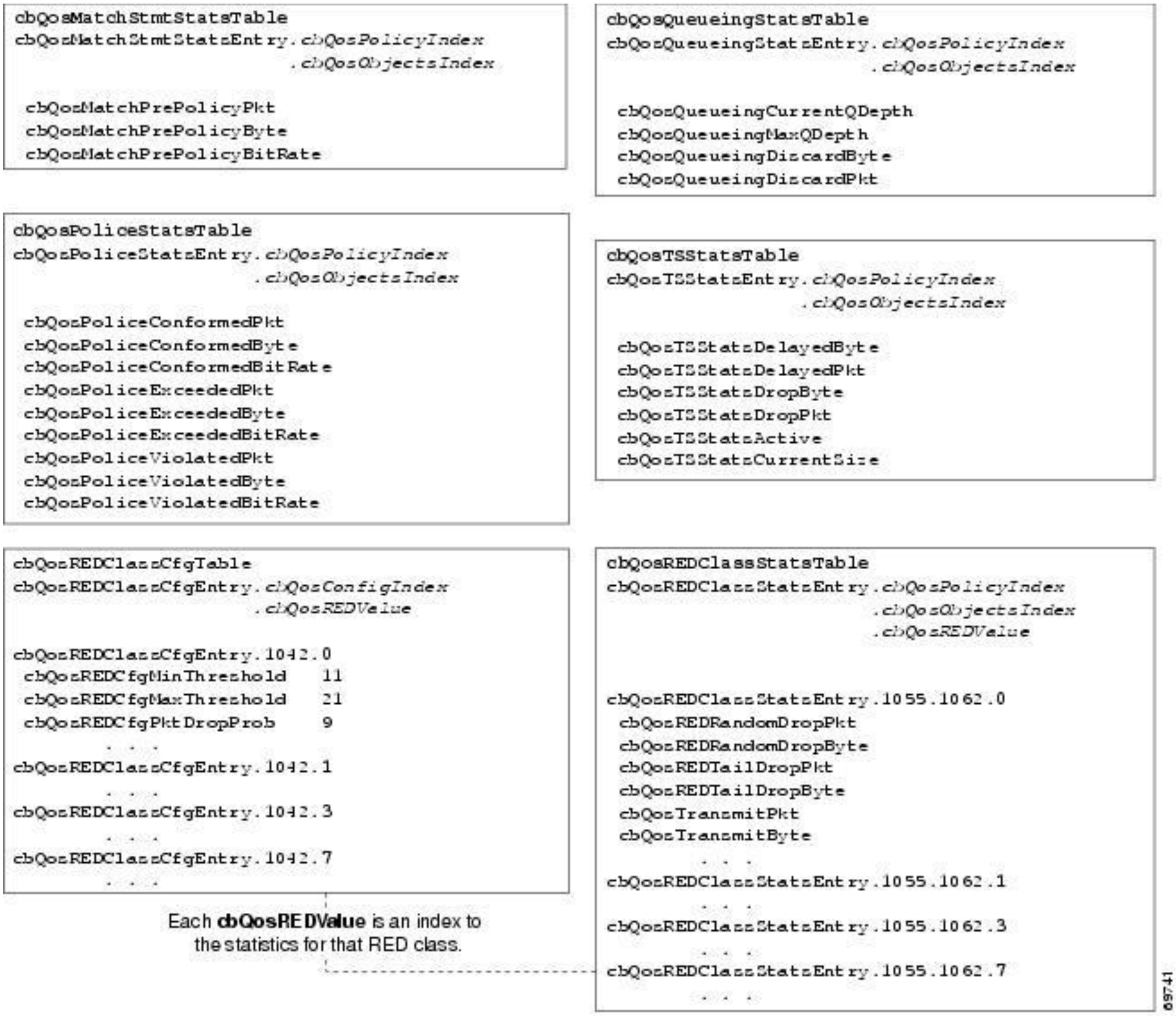
- Figure 1-3 shows the counters in the cbQosCMStatsTable and the indexes for accessing these and other statistics.
- Figure 1-4 shows the counters in cbQosMatchStmtStatsTable, cbQosPoliceStatsTable, cbQosQueueingStatsTable, cbQosTSStatsTable, and cbQosREDClassStatsTable.

For ease-of-use, the following figures show some counters as a single object even though the counter is implemented as three objects. For example, cbQosCMPrePolicyByte is implemented as:

- cbQosCMPrePolicyByteOverflow
- cbQosCMPrePolicyByte
- cbQosCMPrePolicyByte64

Figure 1-3 QoS Class Map Statistics and Indexes





Each **cbQosREDValue** is an index to the statistics for that RED class.

* Counts in cbQosREDClassStatsTable are maintained per class, not cbQosREDValue. All instances of a counter that have the same cbQosREDValue also have the same count.

Figure 1-4 QoS Statistics Tables

Sample QoS Applications

This section presents examples of code showing how to retrieve information from the CISCO-CLASS-BASED-QOS-MIB to use for QoS billing operations. You can use the examples to help you develop billing applications. The topics include:

- Checking Customer Interfaces for Service Policies
- Retrieving QoS Billing Information

Checking Customer Interfaces for Service Policies

This section describes a sample algorithm that checks the CISCO-CLASS-BASED-QOS-MIB for customer interfaces with service policies, and marks those interfaces for further application processing (such as billing for QoS services).

The algorithm uses two SNMP get-next requests for each customer interface. For example, if the router has 2000 customer interfaces, 4000 SNMP get-next requests are required to determine if those interfaces have transmit and receive service policies associated with them.

Note This algorithm is for informational purposes only. Your application needs may be different.

Check the MIB to see which interfaces are associated with a customer. Create a pair of flags to show if a service policy has been associated with the transmit and receive directions of a customer interface. Mark noncustomer interfaces TRUE (so no more processing is required for them).

```
FOR each ifEntry DO
  IF (ifEntry represents a customer interface) THEN
    servicePolicyAssociated[ifIndex].transmit = FALSE;
    servicePolicyAssociated[ifIndex].receive = FALSE;
  ELSE
    servicePolicyAssociated[ifIndex].transmit = TRUE; servicePolicyAssociated[ifIndex].receive = TRUE;
  END-IF
END-FOR
```

Examine the cbQosServicePolicyTable and mark each customer interface that has a service policy attached to it. Also note the direction of the interface.

```
x = 0;
done = FALSE;
WHILE (!done)
  status = snmp-getnext (
      ifIndex = cbQosIfIndex.x,
      direction = cbQosPolicyDirection.x
);
IF (status != 'noError') THEN
   done = TRUE
  ELSE
  x = extract cbQosPolicyIndex from response;
  IF (direction == 'output') THEN
      servicePolicyAssociated[ifIndex].transmit = TRUE;
  ELSE
      servicePolicyAssociated[ifIndex].receive = TRUE;
    END-IF
  END-IF
END-WHILE
```

Manage cases in which a customer interface does not have a service policy attached to it.

```
FOR each ifEntry DO
     IF (!servicePolicyAssociated[ifIndex].transmit) THEN
   Perform processing for customer interface without a transmit service policy.
 END-IF
    IF (!servicePolicyAssociated[ifIndex].receive) THEN
   Perform processing for customer interface without a receive service policy.
  END-IF
END-FOR
```

Retrieving QoS Billing Information

This section describes a sample algorithm that uses the CISCO-CLASS-BASED-QOS-MIB for QoS billing operations. The algorithm periodically retrieves post-policy input and output statistics, combines them, and sends the result to a billing database.

The algorithm uses the following:

One SNMP get request per customer interface–To retrieve the ifAlias.

Two SNMP get-next requests per customer interface–To retrieve service policy indexes.

Two SNMP get-next requests per customer interface for each object in the policy–To retrieve post-policy bytes. For example, if there are 100 interfaces and 10 objects in the policy, the algorithm requires 2000 get-next requests (2 x 100 x 10).

Note This algorithm is for informational purposes only. Your application needs may be different.

Set up customer billing information.

```
FOR each ifEntry DO
 IF (ifEntry represents a customer interface) THEN status = snmp-getnext (id =
    ifAlias.ifIndex); IF (status != 'noError') THEN
Perform error processing.
ELSE
billing[ifIndex].isCustomerInterface = TRUE; billing[ifIndex].customerID = id;
billing[ifIndex].transmit = 0; billing[ifIndex].receive = 0;
   END-IF ELSE
```

```
    billing[ifIndex].isCustomerInterface = FALSE; END-IF
END-FOR
```

Retrieve billing information.

```
x = 0;
done = FALSE; WHILE (!done)
response = snmp-getnext (
ifIndex = cbQosIfIndex.x, direction = cbQosPolicyDirection.x
);
IF (response.status != ‘noError’) THEN done = TRUE
ELSE
x = extract cbQosPolicyIndex from response; IF (direction == ‘output’) THEN
    billing[ifIndex].transmit = GetPostPolicyBytes (x); ELSE
    billing[ifIndex].receive = GetPostPolicyBytes (x); END-IF
END-IF
END-WHILE
```

Determine the number of post-policy bytes for billing purposes.

```
GetPostPolicyBytes (policy) x = policy;
  y = 0; total = 0;
  WHILE (x == policy)
  response = snmp-getnext (type = cbQosObjectsType.x.y); IF (response.status == ‘noError’)
  x = extract cbQosPolicyIndex from response; y = extract cbQosObjectsIndex from response; IF
  (x == policy AND type == ‘classmap’)
  status = snmp-get (bytes = cbQosCMPostPolicyByte64.x.y); IF (status == ‘noError’)
  total += bytes;
      END-IF END-IF
      END-IF END-WHILE
RETURN total;
```

Monitoring Router Interfaces

This section provides information about how to monitor the status of router interfaces to see if there is a problem or a condition that might affect service on the interface. To determine if an interface is Down or experiencing problems, you can:

- see the “Check the Operational and Administrative Status of Interface” section
- see the “Monitor linkDown and linkUp Notifications” section

Check the Operational and Administrative Status of Interface

To check the status of an interface, view the following IF-MIB objects for the interface:

- ifAdminStatus–Administratively configured (desired) state of an interface. Use ifAdminStatus to enable or disable the interface.
- ifOperStatus–Current operational state of an interface.

Monitor linkDown and linkUp Notifications

To determine if an interface has failed, you can monitor linkDown and linkUp notifications for the interface. See the “Enabling Interface linkUp and linkDown Notifications” section for instructions on how to enable the following notifications:

- linkDown–Indicates that an interface failed or is about to fail.
- linkUp–Indicates that an interface is no longer in the down state.

Enabling Interface linkUp and linkDown Notifications

To configure SNMP to send a notification when a router interface changes state to up (ready) or down (not ready), perform the following steps to enable linkUp and linkDown notifications:

Step 1 Issue the following CLI command to enable linkUp and linkDown notifications for most, but not necessarily all, interfaces:

```
Router(config)# snmp-server interface <Interface Type> <Interface Number> notification linkupdown
```

Step 2 View the setting of the ifLinkUpDownTrapEnable object (IF-MIB ifXTable) for each interface to determine if linkUp and linkDown notifications are enabled or disabled for that interface.

Step 3 To enable linkUp and linkDown notifications on an interface, set ifLinkUpDownTrapEnable to enabled (1).

Step 4 To enable the Internet Engineering Task Force (IETF) standard for linkUp and linkDown notifications, issue the **snmp-server trap link ietf** command. (The IETF standard is based on RFC 2233.)

```
Router(config)# snmp-server trap link ietf
```

Step 5 To disable notifications, use the **no** form of the **snmp-server** command.

Billing Customers for Traffic

This section describes how to use SNMP interface counters and QoS data information to determine the amount to bill customers for traffic. It also includes a scenario for demonstrating that a QoS service policy attached to an interface is policing traffic on that interface.

This section contains the following topics:

- Input and Output Interface Counts
- Determining the Amount of Traffic to Bill to a Customer
- Scenario for Demonstrating QoS Traffic Policing

Input and Output Interface Counts

The router maintains information about the number of packets and bytes that are received on an input interface and transmitted on an output interface.

For detailed constraints about IF-MIB counter support, see the “CISCO-MAU-EXT-MIB” section.

Consider the following important information about IF-MIB counter support:

- Unless noted, all IF-MIB counters are supported on the Cisco Carrier Routing System interfaces.
- For IF-MIB high capacity counter support, Cisco conforms to the RFC 2863 standard. The RFC 2863 standard states that for interfaces that operate:
  - At 20 million bits per second or less, 32-bit and packet counters must be supported.
  - Faster than 20 million bits per second and slower than 650 million bits per second, 32-bit packet counters and 64-bit octet counters must be supported.
  - At 650 million bits per second or faster, 64-bit packet counters and 64-bit octet counters must be supported.

- When a QoS service policy is attached to an interface, the router applies the rules of the policy to traffic on the interface and increments the packet and byte counts on the interface.

The following CISCO-CLASS-BASED-QOS-MIB objects provide interface counts:

- cbQosCMDropPkt and cbQosCMDropByte (cbQosCMStatsTable)–Total number of packets and bytes that were dropped as they exceeded the limits set by the service policy. These counts include only those packets and bytes that were dropped as they exceeded service policy limits. The counts do not include packets and bytes dropped for other reasons.
- cbQosPoliceConformedPkt and cbQosPoliceConformedByte (cbQosPoliceStatsTable)–Total number of packets and bytes that conformed to the limits of the service policy and were transmitted.

Determining the Amount of Traffic to Bill to a Customer

Perform the following steps to determine how much traffic on an interface is billable to a particular customer:

Step 1 Determine which service policy on the interface applies to the customer.

Step 2 Determine the index values of the service policy and class map used to define the customer’s traffic. You need this formation in the following steps.

Step 3 Access the cbQosPoliceConformedPkt object (cbQosPoliceStatsTable) for the customer to determine the amount of traffic on the interface that is billable to this customer.

Step 4 (Optional) Access the cbQosCMDropPkt object (cbQosCMStatsTable) for the customer to determine how much of the customer’s traffic was dropped as it exceeded service policy limits.

Scenario for Demonstrating QoS Traffic Policing

This section describes a scenario that demonstrates the use of SNMP QoS statistics to determine how much traffic on an interface is billable to a particular customer. It also shows how packet counts are affected when a service policy is applied to traffic on the interface.

To create the scenario, perform the following steps (each step described in the section below):

Step 1 Create and attach a service policy to an interface.

Step 2 View packet counts before the service policy is applied to traffic on the interface.

Step 3 Issue a **ping** command to generate traffic on the interface. Note that the service policy is applied to the traffic.

Step 4 View packet counts after the service policy is applied to determine how much traffic to bill the customer for:

a. Conformed packets–The number of packets within the range set by the service policy and for which you can charge the customer.

b. Exceeded or dropped packets–The number of packets that were not transmitted because they were outside the range of the service policy. These packets are not billable to the customer.

Note In this scenario, the Cisco Carrier Routing System is used as an interim device (that is, traffic originates elsewhere and is destined for another device).

Service Policy Configuration

The following example uses policy map configuration:

```
policy-map police-out class BGPclass
  police 8000 1000 2000 conform-action transmit exceed-action drop

 interface GigabitEthernet0/1/0/0.10 description VLAN voor klant
  encapsulation dot1Q 10
  ip address 10.0.0.17 255.255.255.248 service-policy output police-out
```

Packet Counts Before the Service Policy Is Applied

The following CLI and SNMP output shows the output traffic for interface before the service policy is applied:

CLI Command Output

```
RSP/0/RSP0/CPU0:ios-xr# show policy-map interface GigabitEthernet0/7/0/0.1
GigabitEthernet0/7/0/0.1 input: policy-police
Class class-out
Classification statistics (packets/bytes) (rate - kbps)
Matched : 0/0 0
Transmitted : Un-determined
Total Dropped : Un-determined
Policing statistics (packets/bytes) (rate - kbps)
Policed(conform) : 0/0 0
Policed(exceed) : 0/0 0
Policed(violate) : 0/0 0
Policed and dropped : 0/0
Class class-default
Classification statistics (packets/bytes) (rate - kbps)
Matched : 0/0 0
Transmitted : Un-determined
Total Dropped : Un-determined
```

SNMP Output

```
ASR 9000# getone -v2c 10.86.0.63 public ifDescr.65
ifDescr.65 = GigabitEthernet0/6/0/0.10
```

Generating Traffic

The following set of **ping** commands generates traffic:

```
ASR 9000# ping
Protocol [ip]:
Target IP address: 10.0.0.18
Repeat count [5]: 99
Datagram size [100]: 1400
Timeout in seconds [2]: 1
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.

Sending 100, 1400-byte ICMP Echos to 10.0.0.18, timeout is 1 seconds:
..!!..!..!..!..!.!.!..!.!..!.!..!.!.!..!.!..!.!..!.!.!..!..!.!.!..!.!..!.!..!.!.!.. !.!..!.!..!.!.!..!.!..!.!.!..!
Success rate is 42 percent (42/100), round-trip min/avg/max = 1/1/1 ms
```

Packet Counts After the Service Policy Is Applied

After you generate traffic using the **ping** command, look at the number of packets that exceeded and conformed to the committed access rate (CAR) set by the **police** command:

- 42 packets conformed to the police rate and were transmitted
- 57 packets exceeded the police rate and were dropped

The following CLI and SNMP output show the counts on the interface after the service policy is applied: (In the output, conformed and exceeded packet counts are shown in boldface.)

CLI Command Output

```
ASR 9000# show policy-map interface g6/0/0.10

GigabitEthernet6/0/0.10

Service-policy output: police-out

    Class-map: BGPclass (match-all)
         198 packets, 281556 bytes
        30 second offered rate 31000 bps, drop rate 11000 bps Match: access-group 101
        Police:
        8000 bps, 1000 limit, 2000 extended limit
        conformed 42 packets, 59892 bytes; action: transmit
        exceeded 57 packets, 81282 bytes; action: drop

    Class-map: class-default (match-any)
         15 packets, 1086 bytes
        30 second offered rate 0 bps, drop rate 0 bps Match: any
        Output queue: 0/8192; 48/59940 packets/bytes output, 0 drops
```

SNMP Output

```
ASR 9000# getmany -v2c 10.86.0.63 public ciscoCBQosMIB
. . .
cbQosCMDropPkt.1143.1145 = 57
. . .
cbQosPoliceConformedPkt.1143.1151 = 42
. . .
```

Using IF-MIB Counters

This section describes the IF-MIB counters and how you can use them on various interfaces and subinterfaces. The subinterface counters are specific to the protocols. This section addresses the IF-MIB counters for ATM interfaces.

The IF-MIB counters are defined considering the lower and upper layers:

- ifInDiscards—Number of inbound packets that were discarded, even though no errors were detected to prevent their being deliverable to a higher-layer protocol. One reason for discarding such a packet is to free up buffer space.
- IfInErrors—Number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol for packet-oriented interfaces.
- ifInUnknownProtos—Number of packets received through the interface that were discarded because of an unknown or unsupported protocol for packet-oriented interfaces.

- ifOutDiscards–Number of outbound packets that were discarded even though no errors were detected to prevent their being transmitted. One reason for discarding such a packet is to free up buffer space.
- ifitOutErrors–Number of outbound packets that could not be transmitted because of errors for packet-oriented interfaces.

The logical flow for counters works as follows:

1. When a packet arrives on an interface, check for the following:
   a. Error in packet–If any errors are detected, increment ifInErrors and drop the packet.
   b. Protocol errors–If any errors are detected, increment ifInUnknownProtos and drop the packet.
   c. Resources (buffers)–If unable to get resources, increment ifInDiscards and drop the packet.
   d. Increment ifInUcastPkts/ifInNUcastPkts and process the packet (at this point, increment ifInOctets with the size of packet).
2. When a packet is to be sent out of an interface:
   a. Increment ifOutUcastPkts/ifOutNUcastPkts (increment ifOutOctets with the size of packet).
   b. Check for errors in packet and if there are any errors in packet, increment ifOutErrors and drop the packet.
   c. Check for resources (buffers) and if you cannot get resources, increment ifOutDiscards and drop the packet.

This following output is an example of IF-MIB entries:

IfXEntry ::=

```
    SEQUENCE {
        ifName                      DisplayString,
        ifInMulticastPkts           Counter32,
        ifInBroadcastPkts           Counter32,
        ifOutMulticastPkts          Counter32,
        ifOutBroadcastPkts          Counter32,
        ifHCInOctets                Counter64,
        ifHCInUcastPkts             Counter64,
        ifHCInMulticastPkts         Counter64,
        ifHCInBroadcastPkts         Counter64,
        ifHCOutOctets               Counter64,
        ifHCOutUcastPkts            Counter64,
        ifHCOutMulticastPkts        Counter64,
        ifHCOutBroadcastPkts        Counter64,
        ifLinkUpDownTrapEnable      INTEGER,
        ifHighSpeed                 Gauge32,
        ifPromiscuousMode           TruthValue,
        ifAlias                     DisplayString,
        ifCounterDiscontinuityTime  TimeStamp
```

Sample Counters

The high capacity counters are 64-bit versions of the basic ifTable counters. They have the same basic semantics as their 32-bit counterparts; their syntax is extended to 64 bits.

This table lists capacity counters object identifiers (OIDs).

Table 1-2 ***Capacity Counters Object Identifiers***

Name	Object Identifier (OID)
ifHCInOctets	::= { ifXEntry 6 }

Table 1-2 ***Capacity Counters Object Identifiers***

Name	Object Identifier (OID)
ifHCInUcastPkts	::= { ifXEntry 7 }
ifHCInMulticastPkts	::= { ifXEntry 8 }
ifHCInBroadcastPkts	::= { ifXEntry 9 }
ifHCOutOctets	::= { ifXEntry 10 }
ifHCOutUcastPkts	::= { ifXEntry 11 }
ifHCOutMulticastPkts	::= { ifXEntry 12 }
ifHCOutBroadcastPkts	::= { ifXEntry 13 }
ifLinkUpDownTrapEnable	::= { ifXEntry 14 }
ifHighSpeed	::= { ifXEntry 15 }
ifPromiscuousMode	::= { ifXEntry 16 }
ifAlias	::= { ifXEntry 18 }
ifCounterDiscontinuityTime	::= { ifXEntry 19 }

QoS MIB Implementation

This appendix provides information about QoS-based features that are implemented on the Cisco Carrier Routing System line cards and what tables and objects in the QoS MIB support these QoS features. The Cisco Carrier Routing System line card families each have a different QoS implementation. Do not assume that the QoS features across line card families are equivalent. Some of the QOS configuration is done at the PFC2 (policy feature card) level and others at the parallel express forwarding (PXF) processor level in each line card.

This appendix contain the following topics:

- Implementing the CISCO-CLASS-BASED-QOS-MIB
- QoS MIB Policy Action Support Matrix

Implementing the CISCO-CLASS-BASED-QOS-MIB

This section describes which objects from the CISCO-CLASS-BASED-QOS-MIB are implemented, which objects are relevant to the features available for the Cisco Carrier Routing System line cards, and which QoS features are supported by each Cisco Carrier Routing System line card.

This table defines the expected values for Policy Actions.

Table 1-1 QoS Policy Action Parameters

Policy Action	Definition	Notes
Bandwidth	A rate limiting function. The difference between the highest and lowest frequencies available for network signals.Bandwidth divides the link bandwidth among different traffic streams into multiple queues.	Must be set before you enable WRED. Aggregate bandwidth rate limits match all of the packets on an interface or subinterface. Granular bandwidth rate limits match a particular type of traffic based on precedence, MAC address, or other parameters.
Priority	Priority queuing allows you to assign a guaranteed minimum bandwidth to one queue to minimize the packet delay variance for delay-sensitive traffic.	A routing feature in which frames in an output queue are prioritized based on various characteristics, such as packet size and interface type.
Shape	A shaper typically delays excess traffic using a buffer or queueing mechanism to hold packets and shape the flow when the data rate of the source is higher than expected. (for example, GTS[1] uses a weighted fair queue to delay packets to shape the flow, and FRTS[2] uses either a PQ[3], a CQ[4], or a FIFO[5] queue for the same, depending on how you configure it.)	Shapers identify traffic descriptor violations.
Police	A policer typically drops traffic. (For example, CAR[6] rate-limiting policer either drops the packet or rewrites its IP precedence, resetting the packet header type of service bits.)	Policing is the process by which the OSR limits the bandwidth consumed by a flow of traffic. Policing can mark or drop traffic.

Table 1-1 QoS Policy Action Parameters

Policy Action	Definition	Notes
Queue limit	Parameter specifies the number of packets held by the queue. It operates on the default packet drop method of congestion management.	A Cisco queuing technique. A flow-based queuing algorithm that creates bit-wise fairness by allowing each queue to be serviced fairly in terms of byte count. For example, if queue 1 has 100-byte packets and queue 2 has 50-byte packets, the WFQ algorithm takes two packets from queue 2 for each one packet from queue 1. This makes service fair for each queue: 100 bytes each time the queue is serviced. WFQ ensures that queues do not starve for bandwidth and that traffic gets predictable service. Low-volume traffic streams–which comprise the majority of traffic–receive increased service, transmitting the same number of bytes as high-volume streams. This behavior results in what appears to be preferential treatment for low-volume traffic, when in actuality it is creating fairness.
Fair queue	Traffic shaping smooths traffic by storing traffic above the configured rate in a queue. When a packet arrives at the interface for transmission, the following happens: ■ If the queue is empty, the arriving packet is processed by the traffic shaper. ■ If possible, the traffic shaper sends the packet. Otherwise, the packet is placed in the queue. ■ If the queue is not empty, the packet is placed in the queue. When there are packets in the queue, the traffic shaper removes the number of packets it can transmit from the queue at each time interval.	A Cisco queuing technique. A flow-based queuing algorithm that creates bit-wise fairness by allowing each queue to be serviced fairly in terms of byte count. For example, if queue 1 has 100-byte packets and queue 2 has 50-byte packets, the WFQ algorithm takes two packets from queue 2 for each one packet from queue 1. This makes service fair for each queue: 100 bytes each time the queue is serviced.
WRED[7]	Action that randomly discards packetsduring IP precedence settings congestion.	Precedence is a value of 0 to 7 where zero is low priority traffic and 7 represents high priority traffic.
Set (precedence)	The IP precedence (QoS) bits in the packet header are rewritten. The packet is then transmitted. You can use this action to either color (set precedence) or recolor (modify existing packet precedence) the packet.	–

[1] GTS = Generic Traffic Shaping
[2] FRTS = Frame Relay Traffic Shaping
[3] PQ = Priority Queue
[4] CQ = Custom Queue
[5] FIFO = first in, first out

6. CAR = Commited Access Rate
7. WRED = Weighted Random Early Detection

Notes About QoS:

- Congestion-management tools include priority queuing (PQ), custom queuing (CQ), weighted fair queuing (WFQ), and class-based weighted fair queuing (CBWFQ).
- Police and shape are traffic regulation mechanisms:
  - Shaping is used to create a traffic flow that limits the full bandwidth potential of the flows. This is used many times to prevent the overflow problem. For instance, many network topologies use Frame Relay in a hub-and-spoke design. In this case, the central site normally has a high-bandwidth link (such as T1), while remote sites have a low-bandwidth link in comparison (such as 384 Kbps). In this case, it is possible for traffic from the central site to overflow the low bandwidth link at the other end. Shaping is a good way to pace traffic closer to 384 Kbps to avoid the overflow of the remote link. Traffic above the configured rate is buffered for transmission later to maintain the rate configured.
  - Policing is similar to shaping, but it differs in one important way; traffic that exceeds the configured rate is not buffered (and normally is discarded).

QoS MIB Policy Action Support Matrix

The tables in this section describe which objects from the CISCO-CLASS-BASED-QOS-MIB are implemented and which ones are relevant to the different features available for the Cisco Carrier Routing System line cards. The tables are divided into objects on the Cisco Carrier Routing System platform that are:

- Supported, implemented, and instrumented (works as defined in the MIB)–Table 1-3
- Not supported or support is limited–Table 1-4

Note The following tables are to be considered examples only as they relate to the ASR 9K.

This table lists the definitions of the values that are returned by objects listed in Table 1-3 and Table 1-4. Policy actions are dependent on return values.

Table 1-2 QoS Table Return Values

Identifier	Definition
Value is **V.**	Returns valid data
Value is **I.** The object is not supported by this platform.	Returns invalid data
Value is a dash (–).	Not instantiated (Does not instantiate [return] any value for this object.)

This table lists QoS MIB table objects that are supported and implemented on the Cisco ASR 9000 Series router platform and the QoS policy actions that these objects support.

Table 1-3 Supported QoS MIB Objects

	Policy Actions								
MIB Tables and Objects	**Band-width**	**Priority**	**Shape**	**Police**	**Queue Limit**	**Fair Queue**	**WRED**	**Set**	**Notes**
cbQosCMStatsTable									
cbQosCMPrePolicy Pkt Overflow	**V**	**V**	**V**	**V**	**V**	**V**	**V**	**V**	The objects listed with a value of **V** are

Table 1-3 Supported QoS MIB Objects

	Policy Actions								
MIB Tables and Objects	**Bandwidth**	**Priority**	**Shape**	**Police**	**Queue Limit**	**Fair Queue**	**WRED**	**Set**	**Notes**
									supported and return valid data.
cbQosCMPrePolicyPkt	V	V	V	V	V	V	V	V	
cbQosCMPrePolicyPkt64	V	V	V	V	V	V	V	V	
cbQosCMPrePolicyByte Overflow	V	V	V	V	V	V	V	V	
cbQosCMPrePolicyByte	V	V	V	V	V	V	V	V	
cbQosCMPrePolicyByte64	V	V	V	V	V	V	V	V	
cbQosCMPrePolicyBitRate	V	V	V	V	V	V	V	V	
cbQosCMPostPolicyByte Overflow	V	V	V	V	V	V	V	V	
cbQosCMPostPolicyByte	V	V	V	V	V	V	V	V	
cbQosCMPostPolicyByte64	V	V	V	V	V	V	V	V	
cbQosCMPostPolicyBit Rate	V	V	V	V	V	V	V	V	
cbQosCMDropPktOverflow	V	V	V	V	V	V	V	V	
cbQosCMDropPkt	V	V	V	V	V	V	V	V	
cbQosCMDropPkt64	V	V	V	V	V	V	V	V	
cbQosCMDropByteOverflow	V	V	V	V	V	V	V	V	
cbQosCMDropByte	V	V	V	V	V	V	V	V	
cbQosCMDropByte64	V	V	V	V	V	V	V	V	
cbQosCMDropBitRate	V	V	V	V	V	V	V	V	
cbQosMatchStmtStatsTable									
cbQosMatchPrePolicyPkt Overflow	I	I	I	V	I	I	I	I	The objects listed with a value of I

Table 1-3 Supported QoS MIB Objects

	Policy Actions								
MIB Tables and Objects	**Band-width**	**Priority**	**Shape**	**Police**	**Queue Limit**	**Fair Queue**	**WRED**	**Set**	**Notes**
									(invalid) are supported but return invalid data for all actions except for Police action (the return data is valid).
cbQosMatchPrePolicyPkt	I	I	I	V	I	I	I	I	
cbQosMatchPrePolicy Pkt64	I	I	I	V	I	I	I	I	
cbQosMatchPrePolicyByte Overflow	I	I	I	V	I	I	I	I	
cbQosMatchPrePolicyByte	I	I	I	V	I	I	I	I	
cbQosMatchPrePolicyBit Rate	I	I	I	V	I	I	I	I	
cbQosMatchPrePolicy Byte64	I	I	I	V	I	I	I	I	
cbQosPoliceStatsTable									
cbQosPoliceConformed PktOverflow	–	–	–	V	–	–	–	–	The objects listed are supported but only return **V** data for Police action.
cbQosPoliceConformedPkt	–	–	–	V	–	–	–	–	
cbQosPoliceConformed Pkt64	–	–	–	V	–	–	–	–	The objects listed are supported but only return **V** data for Police action.
cbQosPoliceConformed ByteOverflow	–	–	–	V	–	–	–	–	
cbQosPoliceConformed Byte	–	–	–	V	–	–	–	–	
cbQosPoliceConformed Byte64	–	–	–	V	–	–	–	–	
cbQosPoliceConformed BitRate	–	–	–	V	–	–	–	–	

Table 1-3 Supported QoS MIB Objects

	Policy Actions								
MIB Tables and Objects	**Bandwidth**	**Priority**	**Shape**	**Police**	**Queue Limit**	**Fair Queue**	**WRED**	**Set**	**Notes**
cbQosPoliceExceededPkt Overflow	–	–	–	**V**	–	–	–	–	
cbQosPoliceExceededPkt	–	–	–	**V**	–	–	–	–	
cbQosPoliceExceeded Pkt64	–	–	–	**V**	–	–	–	–	
cbQosPoliceExceeded ByteOverflow	–	–	–	**V**	–	–	–	–	
cbQosPoliceExceededByte	–	–	–	**V**	–	–	–	–	
cbQosPoliceExceeded Byte64	–	–	–	**V**	–	–	–	–	
cbQosPoliceExceeded BitRate	–	–	–	**V**	–	–	–	–	
cbQosQueueingStatsTable									
cbQosQueueingCurrent QDepth	**V**	**V**	–	–	**V**	**V**	–	–	The objects listed are supported but return valid data only for Bandwidth, Priority, Queue Limit, and Fair Queue.
cbQosQueueingMax QDepth	**V**	**V**	–	–	**V**	**V**	–	–	
cbQosQueueingDiscard ByteOverflow	**V**	**V**	–	–	**V**	**V**	–	–	
cbQosQueueingDiscard Byte	**V**	**V**	–	–	**V**	**V**	–	–	
cbQosQueueingDiscard Byte64	**V**	**V**	–	–	**V**	**V**	–	–	
cbQosQueueingDiscard PktOverflow	**V**	**V**	–	–	**V**	**V**	–	–	
cbQosQueueingDiscardPkt	**V**	**V**	–	–	**V**	**V**	–	–	
cbQosQueueingDiscard Pkt64	**V**	**V**	–	–	**V**	**V**	–	–	
cbQosTSStatsTable									The objects listed are supported but only **V** data for only Shape, Queue Limit,

Table 1-3 ***Supported QoS MIB Objects***

	Policy Actions								
MIB Tables and Objects	**Bandwidth**	**Priority**	**Shape**	**Police**	**Queue Limit**	**Fair Queue**	**WRED**	**Set**	**Notes**
									Fair Queue, and WRED.
cbQosTSStatsDrop Byte Overflow	–	–	**V**	–	**V**	**V**	**V**	–	
cbQosTSStatsDrop Byte	–	–	**V**	–	**V**	**V**	**V**	–	
cbQosTSStatsDrop Byte64	–	–	**V**	–	**V**	**V**	**V**	–	
cbQosTSStatsDrop Pkt Overflow	–	–	**V**	–	**V**	**V**	**V**	–	
cbQosTSStatsDrop Pkt	–	–	**V**	–	**V**	**V**	**V**	–	
cbQosTSStatsDrop Pkt64	–	–	**V**	–	**V**	**V**	**V**	–	
cbQosTSStatsCurrentQSize	–	–	**V**	–	**V**	**V**	**V**	–	
cbQosREDClassStats Table									Not instantiated for Shape even though the CLI shows values for random and tail counters.
cbQosREDRandom Drop PktOverflow	–	–	–	–	–	–	**V**	–	The objects are supported but only **V** data for WRED action only.
cbQosREDRandom DropPkt	–	–	–	–	–	–	**V**	–	
cbQosREDRandom Drop Pkt64	–	–	–	–	–	–	**V**	–	
cbQosREDRandom	–	–	–	–	–	–	**V**	–	
DropByteOverflow									
cbQosREDRandom Drop Byte	–	–	–	–	–	–	**V**	–	
cbQosREDRandom Drop Byte64	–	–	–	–	–	–	**V**	–	
cbQosREDTailDrop Pkt Overflow	–	–	–	–	–	–	**V**	–	
cbQosREDTailDrop Pkt	–	–	–	–	–	–	**V**	–	The objects are supported but only **V** data for WRED action

Table 1-3 Supported QoS MIB Objects

	Policy Actions								
MIB Tables and Objects	**Band-width**	**Priority**	**Shape**	**Police**	**Queue Limit**	**Fair Queue**	**WRED**	**Set**	**Notes**
									only.
cbQosREDTailDrop Pkt64	–	–	–	–	–	–	V	–	
cbQosREDTailDrop Byte Overflow	–	–	–	–	–	–	V	–	
cbQosREDTailDrop Byte	–	–	–	–	–	–	V	–	
cbQosREDTailDrop Byte64	–	–	–	–	–	–	V	–	
cbQosREDTransmit Pkt Overflow	–	–	–	–	–	–	V	–	
cbQosREDTransmit Pkt	–	–	–	–	–	–	V	–	
cbQosREDTransmit Pkt64	–	–	–	–	–	–	V	–	
cbQosREDTransmit Byte Overflow	–	–	–	–	–	–	V	–	
cbQosREDTransmit Byte	–	–	–	–	–	–	V	–	
cbQosREDTransmit Byte64	–	–	–	–	–	–	V	–	

This table lists QoS MIB table objects that are unsupported or have limited support on the Cisco ASR 9000 Series router platform and the QoS policy actions that these objects support.

Table 1-4 QoS MIB Objects–Unsupported or Limited Support

	Policy Actions								
MIB Tables and Objects	**Band-width**	**Priority**	**Shape**	**Police**	**Queue Limit**	**FairQueue**	**WRED**	**Set**	**Notes**
cbQosCMStatsTable									The objects listed are not supported, but return valid data which is always zero(0).
cbQosCMNoBufDropPkt Overflow	V	V	V	V	V	V	V	V	
cbQosCMNoBufDropPkt	V	V	V	V	V	V	V	V	
cbQosCMNoBufDrop Pkt64	V	V	V	V	V	V	V	V	
cbQosPoliceStatsTable									The objects listed are not supported, but

Table 1-4 QoS MIB Objects–Unsupported or Limited Support

	Policy Actions								
MIB Tables and Objects	**Bandwidth**	**Priority**	**Shape**	**Police**	**Queue Limit**	**FairQueue**	**WRED**	**Set**	**Notes**
									return valid data for Police action which is always zero (0).
cbQosPoliceViolatedPktOverflow	–	–	–	V	–	–	–	–	
cbQosPoliceViolatedPkt	–	–	–	V	–	–	–	–	
cbQosPoliceViolatedPkt64	–	–	–	V	–	–	–	–	
cbQosPoliceViolatedByteOverflow	–	–	–	V	–	–	–	–	
cbQosPoliceViolatedByte	–	–	–	V	–	–	–	–	
cbQosPoliceViolatedByte64	–	–	–	V	–	–	–	–	
cbQosPoliceViolatedBitRate	–	–	–	V	–	–	–	–	
cbQosTSStatsTable									The objects listed are not supported but do return valid data which is always zero (0) for Shape, Queue Limit, Fair Queue, and WRED
cbQosTSStatsDelayedByteOverflow	–	–	V		V	V	V	–	
cbQosTSStatsDelayedByte	–	–	V		V	V	V	–	
cbQosTSStatsDelayedByte64	–	–	V		V	V	V	–	
cbQosTSStatsDelayedPktOverflow	–	–	V		V	V	V	–	
cbQosTSStatsDelayedPkt	–	–	V		V	V	V	–	
cbQosTSStatsDelayedPkt64	–	–	V		V	V	V	–	
cbQosTSStatsActive	–	–	I		I	I	I	–	This object is not supported and returns invalid data which is always zero (0) for a truthValue type.

Table 1-4 *QoS MIB Objects–Unsupported or Limited Support*

	Policy Actions								
MIB Tables and Objects	**Bandwidth**	**Priority**	**Shape**	**Police**	**Queue Limit**	**FairQueue**	**WRED**	**Set**	**Notes**
cbQosREDClassStatsTable									The objects listed with a dash (-) are not supported.
cbQosREDECNMarkPktOverflow	–	–	–	–	–	–	–	–	
cbQosREDECNMarkPkt	–	–	–	–	–	–	–	–	
cbQosREDECNMarkPkt64	–	–	–	–	–	–	–	–	
cbQosREDECNMarkByteOverflow	–	–	–	–	–	–	–	–	
cbQosREDECNMarkByte	–	–	–	–	–	–	–	–	
cbQosREDECNMarkByte64	–	–	–	–	–	–	–	–	
cbQosREDMeanQSizeUnits	–	–	–	–	–	–	–	–	
cbQosREDMeanQSize	–	–	–	–	–	–	–	–	
cbQosSetStatsTable									The objects listed with a dash (-) are not supported.
cbQosSetDscpPkt64	–	–	–	–	–	–	–	–	
cbQosSetPrecedencePkt64	–	–	–	–	–	–	–	–	
cbQosSetQosGroupPkt64	–	–	–	–	–	–	–	–	
cbQosSetFrDePkt64	–	–	–	–	–	–	–	–	
cbQosSetAtmClpPkt64	–	–	–	–	–	–	–	–	
cbQosSetL2CosPkt64	–	–	–	–	–	–	–	–	
cbQosSetMplsExpImposition Pkt64	–	–	–	–	–	–	–	–	
cbQosSetDiscardClassPkt64	–	–	–	–	–	–	–	–	
cbQosSetMplsExpTopMost Pkt64	–	–	–	–	–	–	–	–	
cbQosSetSrpPriorityPkt64	–	–	–	–	–	–	–	–	
cbQosSetFrFecnBecnPkt64	–	–	–		–	–	–	–	–
cbQosSetDscpTunnelPkt64	–	–	–		–	–	–	–	–

Table 1-4 ***QoS MIB Objects–Unsupported or Limited Support***

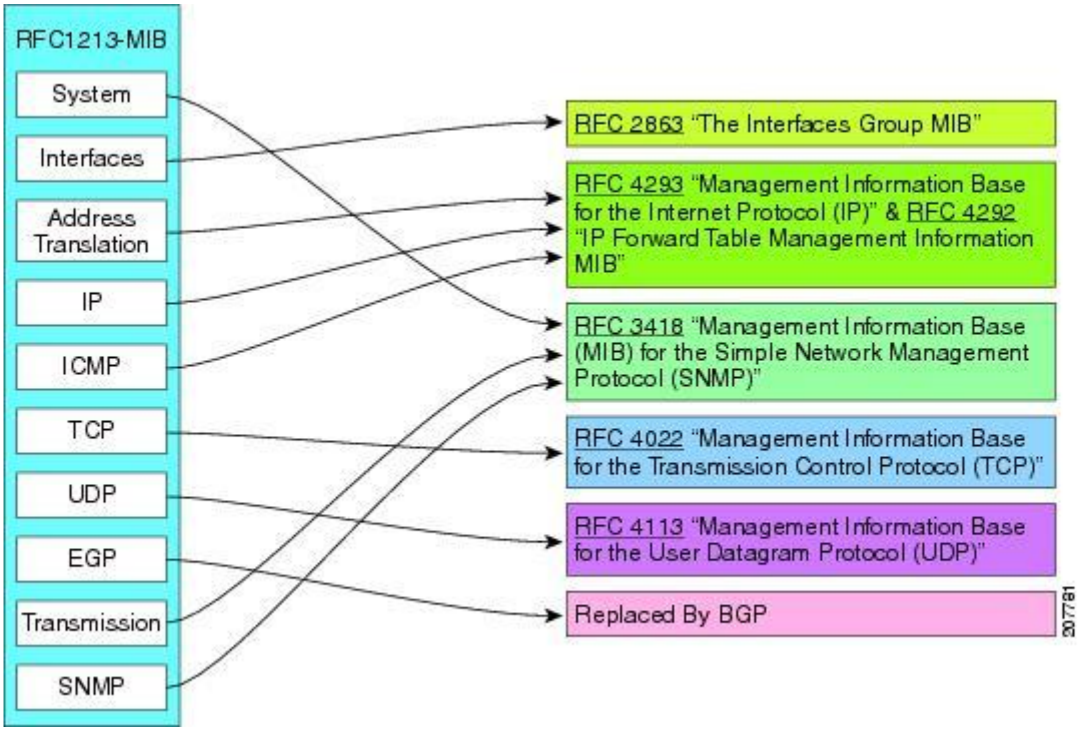
	Policy Actions								
MIB Tables and Objects	**Band-width**	**Priority**	**Shape**	**Police**	**Queue Limit**	**FairQueue**	**WRED**	**Set**	**Notes**
cbQosSetPrecedenceTunnel Pkt64	–	–	–		–	–	–	–	–
cbQosPoliceColorStatsTable									The objects listed with a dash (-) are not supported.
cbQosPoliceCfmColorCfm Pkt64	–	–	–		–	–	–	–	–
cbQosPoliceCfmColorCfm Byte64	–	–	–		–	–	–	–	–
cbQosPoliceCfmColorExd Pkt64	–	–	–		–	–	–	–	–
cbQosPoliceCfmColorExd Byte64	–	–	–		–	–	–	–	–
cbQosPoliceCfmColorVlt Pkt64	–	–	–		–	–	–	–	–
cbQosPoliceCfmColorVlt Byte64	–	–	–		–	–	–	–	–
cbQosPoliceExdColorExdPkt64	–	–	–		–	–	–	–	–
cbQosPoliceExdColorExd Byte64	–	–	–		–	–	–	–	–
cbQosPoliceExdColorVlt Pkt64	–	–	–		–	–	–	–	–
cbQosPoliceExdColorVlt Byte64	–	–	–		–	–	–	–	–
cbQosPoliceVltColorVltPkt64	–	–	–		–	–	–	–	–
cbQosPoliceVltColorVlt Byte64	–	–	–		–	–	–	–	–

RFC 1213

Evolution of RFC 1213

This figure describes the evolution of various groups in RFC 1213. It shows where the information provided by each group in RFC 1213 is available now.

Figure 1-1 RFC 1213 Evolution

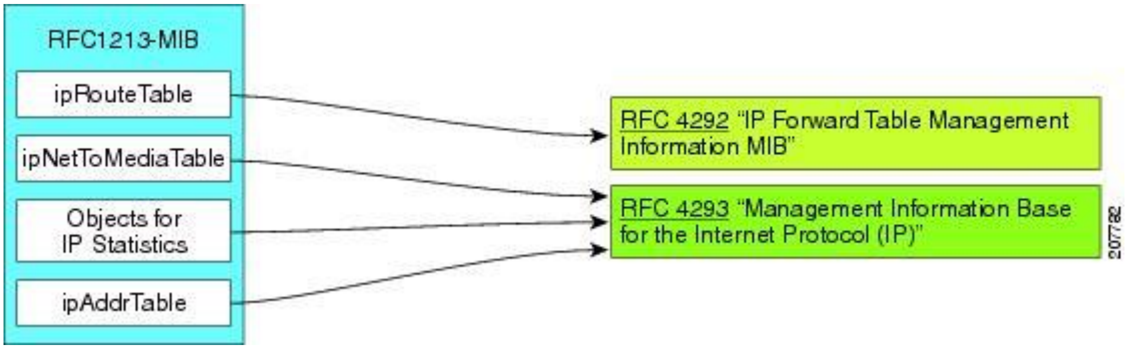

RFC1213-MIB
System
Interfaces
Address Translation
IP
ICMP
TCP
UDP
EGP
Transmission
SNMP
RFC 2863 "The Interfaces Group MIB"
RFC 4293 "Management Information Base for the Internet Protocol (IP)" & RFC 4292 "IP Forward Table Management Information MIB"
RFC 3418 "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)"
RFC 4022 "Management Information Base for the Transmission Control Protocol (TCP)"
RFC 4113 "Management Information Base for the User Datagram Protocol (UDP)"
Replaced By BGP
207781

Evolution of IP Group in RFC 1213

This figure expands the evolution of IP group. It shows the evolution of each table in the group:

- ipRouteTable was replaced by RFC 4292.
- ipNetToMediaTable was replaced by ipNetToPhysicalTable in RFC 4293.
- ipAddrTable was replaced by RFC 4293.

Figure 1-2 Evolution of IP Group in RFC 1213


RFC1213-MIB
ipRouteTable
ipNetToMediaTable
Objects for IP Statistics
ipAddrTable
RFC 4292 "IP Forward Table Management Information MIB"
RFC 4293 "Management Information Base for the Internet Protocol (IP)"
207782

Process Information for SNMP-centric MIBs

Overview of SNMP Framework MIBs

This table is a summary of SNMP Framework MIBs

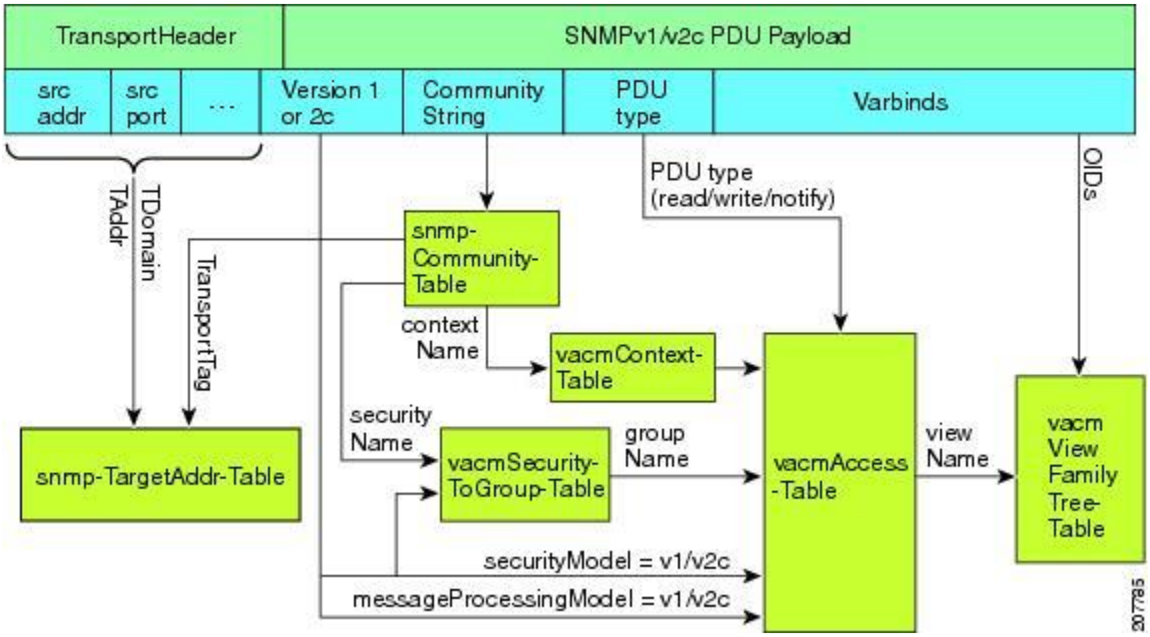
Table 1-1 SNMP Framework MIBs

MIB Module Name	**RFC Number**	**Description**
SNMPv2- MIB	RFC-1907	MIB for SNMPv2
SNMP-FRAMEWORK-MIB	RFC-2571	Architecture for describing SNMP management frameworks
SNMP-MPD-MIB	RFC-2572	Message Processing and Dispatching for SNMP
SNMP-TARGET-MIB	RFC-2573	MIBs for specifying targets of management operations
SNMP-NOTIFICATION-MIB	RFC-2573	MIB for notification filtering
SNMP-USER-BASED-SM-MIB	RFC-2574	USM[1] for version 3
SNMP-VIEW-BASED-ACM-MIB	RFC-2575	VACM[2]
SNMP-COMMUNITY-MIB	RFC-2576	Coexistence between SNMP v1, v2, and v3
NOTIFICATION-LOG-MIB	RFC-2573	MIB for logging SNMP notifications
CISCO-BULK-FILE-MIB	—	MIB module for creating and deleting bulk files of SNMP data for file transfer
CISCO-FTP-CLIENT-MIB	—	MIB module for invoking internet FTP operations for network management processes

[1] USM = User-based security model

[2] VACM = View-based access control model

SNMP Message Processing



SNMPv1 and v2c Coexistence Message Processing

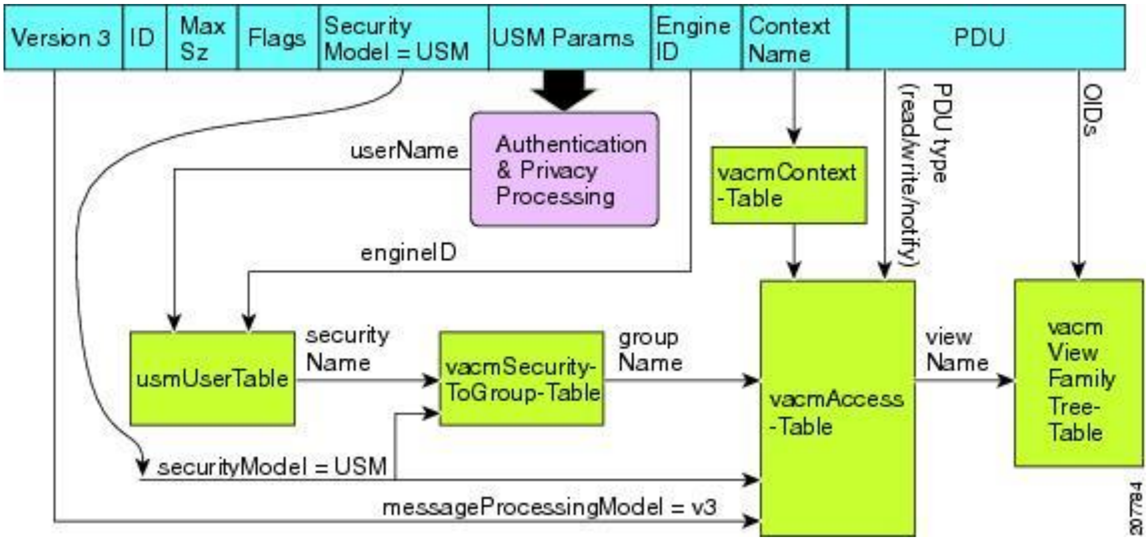
This figure describes SNMPv1 and v2c Coexistence Message Processing.

Figure 1-1 SNMPv1 and v2c Coexistence Message Processing

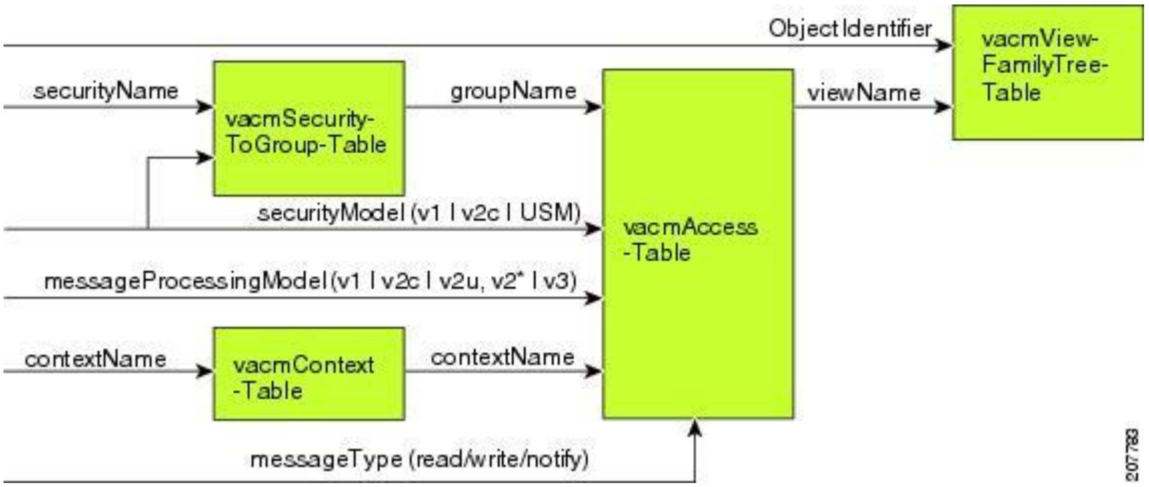
SNMPv3 Message Processing

This figure shows SNMPv3 PDU (Protocol Data Unit) with USM (user-based security model).

Figure 1-2 SNMPv3 Message Processing



SNMPv3 View-Based Access Control Model


ObjectIdentifier
vacmView-FamilyTree-Table
securityName
vacmSecurity-ToGroup-Table
groupName
viewName
securityModel (v1 I v2c I USM)
vacmAccess-Table
messageProcessingModel (v1 I v2c I v2u, v2* I v3)
contextName
vacmContext-Table
contextName
messageType (read/write/notify)
207783

This figure shows the SNMPv3 View-Based Access Control Model.

Figure 1-3 SNMPv3 View-Based Access Control Model

SNMPv1/v2 Community Configuration to Tables Mappings

SNMPv1/v2 community config

Example configuration command used for the following tables: **snmp-server view** *tim2* **RW**

Note The first row of Table 1-2 is a table index.

This table is a summary of SNMPv1/v2 community config for SNMP-COMMUNITY-MIB

Table 1-2 SNMP-COMMUNITY-MIB: snmpCommunityTable

SNMP-COMMUNITY-MIB:: snmpCommunityTable (causes row creation)	
Configuration	**Sample Output**
snmpCommunityName	“tim2”
snmpCommunitySecurityName	“tim2”
snmpCommunityContextEngineID	<localEngineID>
snmpCommunityContextName	“”
snmpCommunityTransportTag	“”

Note The first four rows of Table 1-3 are table indexes.

This table is a summary of SNMPv1/v2 community config for SNMP-VACM-MIB

Table 1-3 SNMP-VACM-MIB: vacmAccessEntry

SNMP-VACM-MIB:: vacmAccessEntry (causes row creation)	
Configuration	**Sample Output**
vacmGroupName	“test-group”
vacmContextPrefix	“”
vacmSecurityModel	1 and 2 (v1 and v2)
vacmSecurityLevel	1 (noAuthNoPriv)
vacmAccessReadViewName	“v1default”
vacmAccessWriteViewName	“v1default”
vacmAccessNotifyViewName	“v1default”

This table is a summary of SNMPv1/v2 community config for SNMP-VACM-MIB

Note The first two rows of Table 1-4 are table indexes.

Table 1-4 SNMP-VACM-MIB: vacmSecurityToGroupEntry

SNMP-VACM-MIB:: vacmSecuritytoGroupEntry (causes row creation)	
Configuration	**Sample Output**
vacmSecurityModel	1 and 2 (v1 and v2)
vacmSecurityName	“tim2”
vacmGroupName	“tim2”

SNMPv3 Configuration to Tables Mappings

SNMPv3 user config

Example configuration command used for the following tables: **snmp-server user** *tim test-group v3*

Note The first two rows of Table 1-5 and Table 1-6 are table indexes.

This table is a summary of SNMPv3 user config for SNMP-USM-MIB

Table 1-5 SNMP-USM-MIB: usmUserEntry

SNMP-USM-MIB:: usmUserEntry (causes row creation)	
Configuration	**Sample Output**
usmUserEngineID	<localEngineID>
usmUserName	“tim”
usmUserSecurityName	“tim”
usmUserPrivProtocol	usmNoPrivProtocol
usmUserAuthProtocol	usmNoAuthProtocol

This table is a summary of SNMPv3 user config for SNMP-VACM-MIB

Table 1-6 SNMP-VACM-MIB: vacmSecurityToGrouptEntry

SNMP-VACM-MIB:: vacmSecurityToGroupEntry (causes row creation)	
Configuration	**Sample Output**
vacmSecurityModel	3
vacmSecurityName	“tim”
vacmGroupName	“test-group”

SNMPv3 group config

Example configuration command used for Table 1-7: **snmp-server group** *test_group v3 noauth* **notify** *test-view* **read** *test-view* **write** *test-view*

Note The first four rows of Table 1-7 are table indexes.

This table is a summary of SNMPv3 group config for SNMP-VACM-MIB

Table 1-7 SNMP-VACM-MIB: vacmAccessEntry

SNMP-VACM-MIB:: vacmAccessEntry (causes row creation)	
Configuration	**Sample Output**
vacmGroupName	“test-group”
vacmConextPrefix	“”
vacmSecurityModel	3 (v3)
vacmSecurityLevel	1 (noAuthNoPriv)
vacmAccessReadViewName	“test-view”
vacmAccessWriteViewName	“test-view”
vacmAccessNotifyViewName	“”

SNMPv3 view config

Example configuration command used for Table 1-8: **snmp-server view** *test-view internet* **included**

Note The first two rows of Table 1-8 are table indexes.

This table is a summary of SNMPv3 group config for SNMP-USM-MIB

Table 1-8 SNMP-USM-MIB: vacmViewTreeFamilyEntry

SNMP-VACM-MIB:: vacmViewTreeFamilyEntry (causes row creation)	
Configuration	**Sample Output**
vacmViewTreeFamilyViewName	“test-view”
vacmViewTreeFamilySubtree	.1.3.6.1 (OID = internet)
vacmViewTreeFamilyMask	“”
vacmViewTreeFamilyType	included

IOS XR SNMP Best Practices

Simple Network Management Protocol (SNMP) is the most common network management protocol in the routing industry. This chapter describes best practices to be adopted by an Operations Support System (OSS) for optimized use of the IOS-XR SNMP protocol.

Note OSS platform tuning and Data Communication Network (DCN) considerations are outside the scope of this document.

Overview

The implementation of management infrastructure for IOS XR network elements usually makes use of SNMP as a first line tool, in particular for fault management and statistics reports.

This chapter provides guidelines on how to interface the SNMP Agent of the IOS XR network equipment in different ways. The goals are to:

- Prevent SNMP congestion
- Achieve a better response time
- Avoid losing traps

Timeouts and Retries

The access to SNMP tables or variables cannot be instantaneous, and given the UDP nature of the protocol, the OSS cannot rely on a fast response time. In addition, SNMP is a non confirmed protocol and therefore, the OSS has to be tuned to retry after a GET operation times out.

Timeouts

The timeout to be tuned on the SNMP management application depends on various factors:

- Response time of the Network Element (NE), which largely depends on the presence of data in cache vs. dynamic retrieve
- CN delay, usually shorter than the above, but sometimes not negligible because of congestion
- Number of SNMP management applications polling the same NE

While a timeout of 1 second can be considered in a lab, in a real application at least 3.5 seconds is preferred. The best approach is a dynamic timeout, where the OSS automatically adjusts the timeout upon different poll retries. This approach, where present, optimizes response time and takes into account all of the above factors.

Where dynamic tuning is not available at OSS level, a row formula for a timeout can be the following:

N x TRT + DCNd

Where:

- N—Number of management applications (for example, SNMP Managers) polling an SNMP agent at the same time. Maximum recommended is 5
- RT—Regular timeout for a single management application. As stated above, 3.5 seconds is recommended
- DCNd—DCN delay for the given OSS or NE pair

Timeout Recommendations

1. Use dynamic timeout if available
2. Use a 3.5 second timeout if dynamic timeout is not available
3. Have no more than 5 management applications polling SNMP at the same time.

Retries

When a poll timeout expires, the typical OSS performs a certain number of retries before declaring an object as *unreachable*. For timeout fine tuning, the number of retries should also be tuned depending on the same factors. Dynamic adjustment is recommended when present, otherwise heuristic calculations need to be used.

As a general consideration, a retry should be done with the same SNMP request-id and the same IP source port. This takes the first available response and also helps the SNMP agent to drop multiple retries directed to the same object.

Retry Recommendations

1. Use dynamic retry if available
2. Calculate static retry based on factors listed, if dynamic retry is not available
3. On all retries, use the same SNMP request-id and IP source port

Tables

A typical MIB table is defined as SEQUENCE of SEQUENCE in MIB syntax, where each SEQUENCE contains a set of MIB objects. An instance of a SEQUENCE represents a row, and all instances of the MIB objects represent a column in a conceptual MIB table. The way in which a table is traversed significantly impacts the response time. Many considerations need to be examined when accessing a table and these depend on the table itself. First, the way in which the table is traversed may affect the response time, and secondly, the nature of the table itself suggests some good practices.

Accessing Tables

Data in a table can be retrieved by doing a sequence of SNMP GET-NEXT requests by one or more GET-BULK requests. Irrespective of the type of request, traversal in a table is essentially fetching the ‘next’ instance of an object. This is possible because MIB tables are expected to be sorted lexicographically based on one or more indices.

Each row in a conceptual MIB table contains attributes or statistics for a specific entity. What this entity represents is purely based on MIB definition. Most of the MIB implementations act as front end for a specific feature. The set of data that is required could be deep within a process or hardware counter that implements the actual feature. In a distributed environment, that data could even be in a line card. Various data pertaining to a specific entity is often kept together as a group and thus the cost of fetching this group is often the same as getting one element from this group—because of inter-process communication overhead (here the cost is in terms of CPU and time).

It is therefore much more efficient to access all objects for an instance or entity than accessing them one by one. This can be considered as row-by-row traversal. Achieving this with GET-NEXT or GET-BULK operation means specifying all required objects of a table in the same request. This approach is usually better, except in the case of sparse tables. For more information, see the “Sparse Tables” section.

This table is an example of SNMP best practices when accessing a table.

Note In Table 1-1 each line under ‘Column-wise’ represents the output of a single SNMP request and in the ‘Row-wise’ column not all of the objects are listed (for brevity each SNMP request includes only three objects).

Table 1-1 ***Accessing the ifTable of IF-MIB***

Column-wise	Row-wise
bash-2.03$ getmany -v2c 12.25.26.10 public ifTable	bash-2.03$ bash-2.03$ getmany -v2c 12.25.26.10 public ifDescr ifType ifAdminStatus
ifIndex.1 = 1	ifDescr.1 = MgmtEth0/RP1/CPU0/0
ifIndex.2 = 2	ifType.1 = ethernetCsmacd(6)
ifIndex.3 = 3	ifAdminStatus.1 = up(1)
ifIndex.4 = 4	ifDescr.2 = Null0
ifIndex.5 = 5	ifType.2 = other(1)
ifIndex.6 = 6	ifAdminStatus.2 = up(1)
ifIndex.7 = 7	ifDescr.3 = SONET0/2/0/0
	ifType.3 = sonet(39)
	ifAdminStatus.3 = up(1)
	ifDescr.4 = POS0/2/0/0
	ifType.4 = pos(171)
	ifAdminStatus.4 = down(2)

Sparse Tables

A *sparse table* is a table where only a subset of columnar objects is instantiated for each row. In MIB tables this occurs when certain objects are not instantiated for certain rows, because it is either not applicable or not available at that time (for example, in the ifTable, some of the counters are not applicable for tunnel interfaces). Sparse MIB tables are potential sources of low performance during table traversal, especially with row-wise access.

This table is a snapshot from the ifTable of IF-MIB showing a sparse entry for SONET0/2/0/0.

Table 1-2 ***Example of sparse entry from IF-MIB***

ifDescr	isInOctets
MgmtEth0/RP1/CPU0/0	5036844
SONET0/2/0/0	
POS0/2/0/0	1024

During row-wise traversal each request would contain objects that are of interest to the OSS. MIB implementation processes each of the requests one by one. For each of these objects it needs to identify the ‘next’ instance and the value for this instance. Identifying “next” instance could be a CPU intensive operation as it might involve sorting or searching internal data structures or even external data structures in distributed environments. As an optimization, this “next” instance identification needs to be done only once per request containing multiple objects from the same table. This optimization would break if one sof the objects is not applicable or not available for already identified “next” instance. This will result in further searching until the system identifies a new “next” instance for that object.

Note The best approach is to avoid these sparse objects during row-wise traversal of MIB tables by avoiding GET-NEXT over unsupported objects.

A row-wise GET on some of the selected objects from IF-MIB (ifTable) is shown below. Each request contains two objects. Third response onwards contains a “jump” over sparse object and this “jump” requires additional searching every time.
bash-2.03$ getmany -v2c 12.25.26.10 public ifDescr ifInOctets

```
ifDescr.1 = MgmtEth0/RP1/CPU0/0
ifInOctets.1 = 5072278
ifDescr.2 = Null0
ifInOctets.2 = 0
ifDescr.3 = SONET0/2/0/0
ifInOctets.4 = 0
ifDescr.4 = POS0/2/0/0
ifInOctets.6 = 0
 ifDescr.5 = SONET0/2/0/1
ifInOctets.8 = 0
```

Requests Addressed to Interleaved Objects

Accessing all the required objects of a row in a single request is much more efficient than making multiple queries for each. Single requests can contain objects for multiple rows also. It is important to note that the request should contain objects with the same instance in sequence. Interleaving objects with different instances would result in issues similar to sparse tables and multiple retrieval of same information from feature related process.

The example below shows an SNMP-GET request performed on IF-MIB (ifTable). The request contains objects from two different interfaces. Object instances are interleaved in the request.

```
bash-2.03$ getone -v2c 12.25.26.10 public ifDescr.1 ifType.2 ifType.1 ifDescr.2 ifDescr.1 = MgmtEth0/RP1/CPU0/0
ifType.2 = other(1)
ifType.1 = ethernetCsmacd(6) ifDescr.2 = Null0
bash-2.03$
```

Rearranging this request with consecutive objects for the same instance would be more efficient in terms of processing.

```
bash-2.03$ getone -v2c 12.25.26.10 public ifDescr.1 ifType.1 ifType.2 ifDescr.2 ifDescr.1 = MgmtEth0/RP1/CPU0/0
ifType.1 = ethernetCsmacd(6) ifType.2 = other(1)
ifDescr.2 = Null0
bash-2.03$
```

Large Tables

Any MIB table that is expected to contain 100 entries is considered a large table. IF-MIB, IPFORWARD-MIB, and IP-MIB have examples of large tables. Traversing these tables should be performed in a more intelligent way as it can consume a lot of resources. OSSs should make use of any existing additional objects which provide overall information about the table, for example, an object describing number of entries or last modified time. Last modified time could be used to identify whether there have been any changes to the table since last retrieval. An object representing number of entries could be used to split the retrieval into multiple sets of smaller discoveries separated by time. For example:

- IP-FORWARD-MIB::ipCidrRouteNumber
- IP-MIB::ipv4InterfaceTableLastChange
- ENTITY-MIB::entLastChangeTime

Static Data

Some tables actually contain data which do not change during a management session. Data which are almost static should be retrieved at the beginning once and then no longer accessed. In other words an SNMP agent should not be used as a repository to be polled for constant data. Of course a smart OSS should know when it is time to fetch the data again, if some event takes place of OIR. If these data are fetched once only, this saves CPU time for other requests as well as useless DCN load.

- ENTITY-MIB::entPhysicalTable

Use of SNMP views

Access to MIB tables or objects can be optimized with the use of appropriate SNMP view configurations. This is useful when an OSS makes frequent retrieval of very large tables and there is no direct control on it. To achieve this, an SNMP view has to be configured by excluding respective MIB sub-trees and attaching this view to the SNMP community name or group name being used. This approach can also be applied in cases where certain MIB implementations being polled have to be prevented (for example, very slow response or impacting the performance of other MIBs). The following shows a view configuration and associated community configuration which prevents IP-FORWARDMIB (ipCidrRouteTable) and IP-MIB (ipNetToMediaTable) being polled. SNMP query made with community string “mycom” will not access these two tables.

- snmp-server view cutdown 1.3.6.1.2.1.4.22 excluded
- snmp-server view cutdown 1.3.6.1.2.1.4.24.4 excluded
- snmp-server community mycom view cutdown

Cut down views can also be created by excluding everything and including only what is required by OSSs.

Table Access Recommendations

1. In a sparse table do specific GETs and avoid using GET-NEXT over unsupported objects.
2. In non-sparse tables, use GET or GET-BULK to acquire all related objects in one request instead of using GET-NEXT to walk a table.
3. Consider if objects are interleaved and use this information to efficiently make GET requests.
4. When possible, use objects which provide information about large tables instead of accessing the whole table.
5. Do not retrieve nearly static data more often than absolutely necessary (for example, at the start of an OSS session).
6. Use SNMP views to carefully select what information is retrieved.

Multiple OSS

Parallel access of the same MIB from multiple polling stations can lead to slower response and higher CPU utilization. This is evident especially in the case of large tables, where multiple stations access different part of the same MIB table. In distributed environments most of the MIB implementations would have some kind of look-ahead caching done. This holds well if requests fall in this cache. This would lead to performance issues because of repeated cache misses OTHERs.

Multiple OSS recommendation: Avoid having multiple OSSs polling the same MIB–instead share the data between the OSSs.

MIB Specific Functionality

Some MIBs provide key features that permit improved NMS/OSS operation.

One such feature is IF-MIB ifindex persistence feature (also relevant for other MIBs that use the ifIndex such as Etherlike-MIB). This feature permits ifindices to remain constant across reboots, which prevents the need for the NMS/OSS to perform interface rediscovery after a reboot. Another similar feature is the ENTITY-MIB entphyscialindex persistence (also relevant for MIBs that extend the Entity-MIB such as the CISCO-ENTITY-FRUCONTROL-MIB). Similar to ifindex persistence, this feature permits entphysical indices to persist across reboots. Finally, the CISCO-CLASS-BASED-QOS-MIB also has an index persistence function, which prevents the need to rediscover QoS indices after reboot. To obtain information on the configuration of these features see the IOS-XR System Management Command Reference documents at:http://www.cisco.com/en/US/products/ps5845/prod_command_reference_list.html

General Performance Considerations and Tunable Parameters

Performance tuning and considerations for SNMP stem from two different perspectives:

- Desire to adjust/improve the performance of SNMP in specific operations
- Desire to limit impact of SNMP performance on overall device operations

IOS-XR provides a number of options to assist in both of these areas. IOS-XR supports both in band and out of band SNMP operation. In band processing can be controlled via LPTS (control plane policing) in the same way as other control plane operations. Both in band and out of band operation can further be tuned by the following mechanisms:

- Request throttling [snmp-server throttle-time]
- Queue length throttling (trap only) [snmp-server queue-length]

Note This only affects the outgoing trap queue, not the incoming request queue.

- SNMP Overload Control [snmp-server overload]
- Protection of critical operations (such as routing convergence) from SNMP CPU overuse.

To obtain information on the configuration of these features see the IOS-XR System Management Command Reference documents at:http://www.cisco.com/en/US/products/ps5845/prod_command_reference_list.html

Note that many SNMP performance considerations are related to specific MIB implementations, but the above system wide settings do have some affect on performance.

MIB Specific Performance Considerations and Tunable Parameters

MIB polling performance is frequently associated with the implementation of a specific MIB and there are some tunable parameters for specific MIBs. For the IF-MIB, caching support is configured using the **snmp-server ifmib stats cache** command.

Note This command only affects the SNMP specific interface statistic operation, the behavior of the IOS-XR internal statistics collection mechanism in IOS-XR is not controllable from SNMP.

The CISCO-CLASS-BASED-QOS-MIB also has caching support, this is configured via the **snmp-server mibs cbqosmib cache** command. The obvious tradeoff in the use of these caches is faster performance versus most recent data. To obtain information on the configuration of these features see the IOS-XR System Management Command Reference documents at:
http://www.cisco.com/en/US/products/ps5845/prod_command_reference_list.html