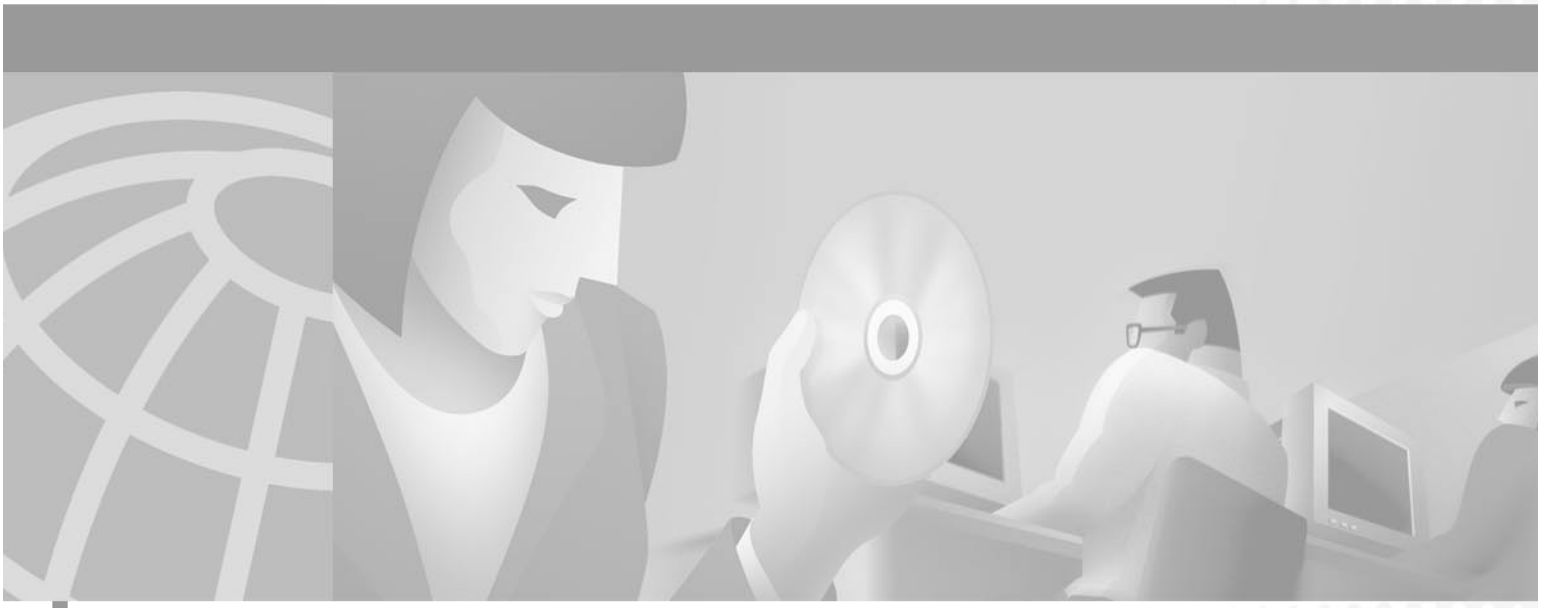


Cisco MGX 8850 SNMP Reference

Release 2.1
December 2001

Corporate 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: 408 526-4100

Customer Order Number: DOC-7812562=
Text Part Number: 78-12562-01 Rev. D0

Cisco Reader Comment Card

General Information

1 Years of networking experience: _____ Years of experience with Cisco products: _____

2 I have these network types: ☐ LAN ☐ Backbone ☐ WAN
☐ Other: _____

3 I have these Cisco products: ☐ Switches ☐ Routers
☐ Other (specify models): _____

4 I perform these types of tasks: ☐ H/W installation and/or maintenance ☐ S/W configuration
☐ Network management ☐ Other: _____

5 I use these types of documentation: ☐ H/W installation ☐ H/W configuration ☐ S/W configuration
☐ Command reference ☐ Quick reference ☐ Release notes ☐ Online help
☐ Other: _____

6 I access this information through: ____% Cisco.com ____% CD-ROM
____% Printed docs ____% Other: _____

7 I prefer this access method: _____

8 I use the following three product features the most:

Document Information

Document Title: *Cisco MGX 8850 SNMP Reference*

Part Number: 78-12562-01 Rev. D0 S/W Release (if applicable): Release 2.1

On a scale of 1–5 (5 being the best), please let us know how we rate in the following areas:

_____ The document is written at my technical level of understanding.

_____ The information is accurate.

_____ The document is complete.

_____ The information I wanted was easy to find.

_____ The information is well organized.

_____ The information I found was useful to my job.

Please comment on our lowest scores:

Mailing Information

Company Name Date

Contact Name Job Title

Mailing Address

City State/Province ZIP/Postal Code

Country Phone ( ) Extension

Fax ( ) E-mail

Can we contact you further concerning our documentation? ☐ Yes ☐ No

You can also send us your comments by e-mail to **bug-doc@cisco.com**, or by fax to **408-527-8089**.

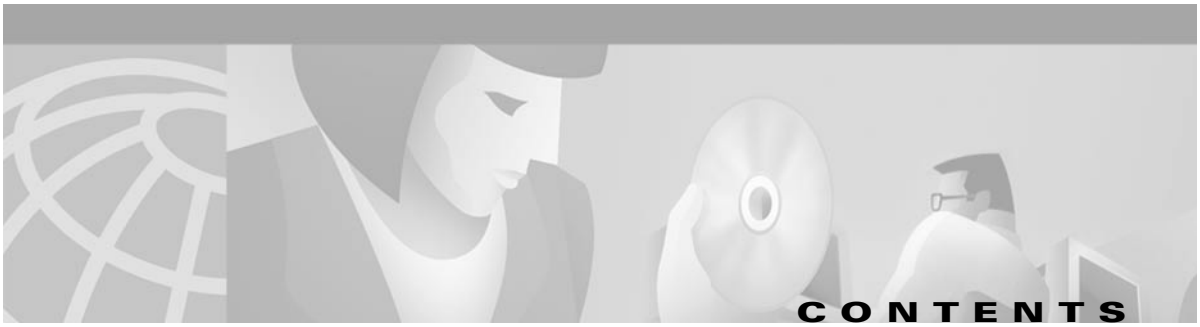
NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

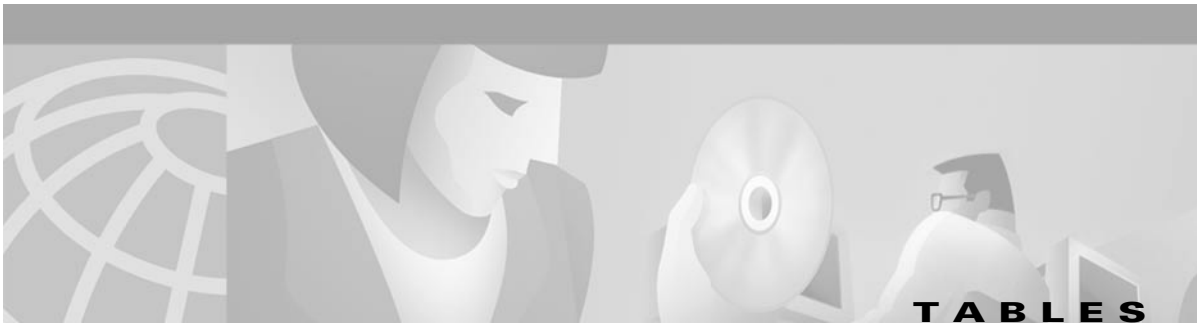
FIRST-CLASS MAIL PERMIT NO. 4631 SAN JOSE CA

POSTAGE WILL BE PAID BY ADDRESSEE

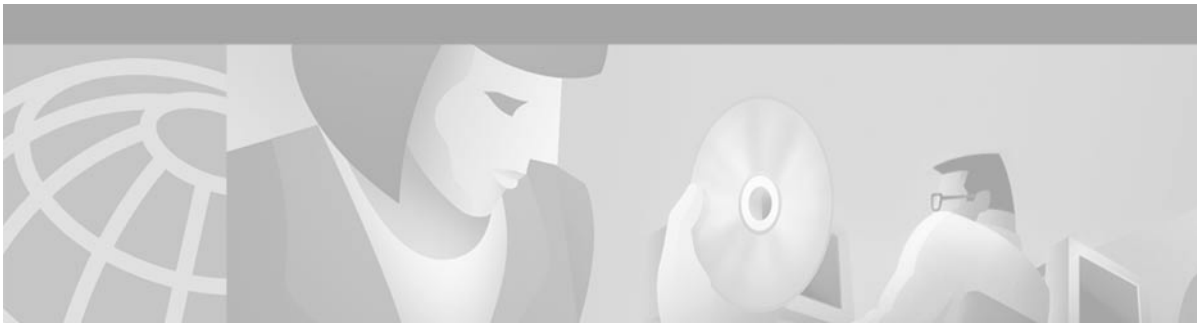
ATTN DOCUMENT RESOURCE CONNECTION
CISCO SYSTEMS INC
170 WEST TASMAN DRIVE
SAN JOSE CA 95134-9883



CONTENTS



TABLES



About This Guide

This chapter discusses the objectives, audience, organization, and conventions of the *Cisco* MGX 8850 *SNMP Reference*.

Objectives

This guide provides reference material for those who want to use the AXSM, AXSM-E, PXM, and RPM modules for MGX 8850 in regards to the Management Information Base (MIB).

Audience

This publication is intended for users who intend to develop applications such as provisioning, or to create or modify an existing Network Management System (NMS) for the switch.

Organization

This guide is organized as follows:

- Chapter 1, “MIBs Supported,” describes the existing MIBs that apply to AXSM, PXM, and RPM.
- Chapter 2, “Standard MIB Objects,” describes the standard MIB objects that are common to AXSM, PXM, and RPM.
- Chapter 3, “Cisco Enterprise MIB Objects,” describes the common MIB objects used for the Cisco enterprise for both AXSM, PXM, and RPM.
- Chapter 4, “PXM MIB Objects,” describes the MIB objects used for PXM.
- Chapter 5, “PNNI MIB Objects,” describes the MIB objects used for PNNI.
- Chapter 6, “RPM MIB Objects,” describes the MIB objects used for RPM.
- Chapter 7, “Traps and Alarms Supported,” describes the traps and alarms used for AXSM, PXM, and RPM.
- Chapter 8, “Agent Capabilities Supported,” describes the agent capabilities used for AXSM, PXM, and RPM service modules.
- Chapter 9, “Statistics Supported,” describes the statistics supported by the AXSM MIBs.

Related Documentation

The following Cisco publications contain additional information related to the operation of this product and associated equipment in a Cisco WAN switching network.

Cisco WAN Manager Release 10.5 Documentation

The product documentation for the Cisco WAN Manager (CWM) network management system for Release 10.5 is listed in Table 1.

Table 1 Cisco WAN Manager Release 10.5 Documentation

Title	Description
Cisco WAN Manager Installation Guide for Solaris, Release 10.5 DOC-7812948=	Provides procedures for installing Release 10 of the CWM network management system and Release 5.3 of CiscoView.
Cisco WAN Manager User's Guide, Release 10.5 DOC-7812945=	Describes how to use the CWM Release 10 software which consists of user applications and tools for network management, connection management, network configuration, statistics collection, and security management.
Cisco WAN Manager SNMP Service Agent, Release 10.5 DOC-7812947=	Provides information about the CWM Simple Network Management Protocol Service Agent, an optional adjunct to CWM used for managing Cisco WAN switches using SNMP.
Cisco WAN Manager Database Interface Guide, Release 10.5 DOC-7812944=	Provides information about accessing the CWM Informix OnLine database that is used to store information about the network elements.

Table 2 WAN CiscoView Release 10 Documentation

Title	Description
WAN CiscoView Release 3 for the MGX 8850 Edge Switch, Release 1 DOC-7811242=	Provides instructions for using this network management software application that allows you to perform minor configuration and troubleshooting tasks.
WAN CiscoView Release 3 for the MGX 8250 Edge Concentrator, Release 1 DOC-7811241=	Provides instructions for using this network management software application that allows you to perform minor configuration and troubleshooting tasks.
WAN CiscoView Release 3 for the MGX 8230 Multiservice Gateway, Release 1 DOC-7810926=	Provides instructions for using this network management software application that allows you to perform minor configuration and troubleshooting tasks.

Cisco MGX 8850 Release 2.1 Documentation

The product documentation for the installation and operation of the MGX 8850 Release 2.1 switch is listed in Table 3.

Table 3 Cisco MGX 8850 Switch Release 2.1 Documentation

Title	Description
Cisco MGX 8850 Routing Switch Hardware Installation Guide, Release 2.1 DOC-7812561=	Describes how to install the MGX 8850 routing switch. It explains what the switch does, and covers site preparation, grounding, safety, card installation, and cabling.
Cisco MGX 8850 Switch Command Reference, Release 2.1 DOC-7812563=	Describes how to use the commands that are available in the CLI[1]of the MGX 8850 switch.
Cisco MGX 8850 Switch Software Configuration Guide, Release 2.1 DOC-7812551=	Describes how to configure the MGX 8850 switch to operate as ATM edge and core switch. This guide also provides some operation and maintenance procedures.
Cisco MGX 8850 SNMP Reference, Release 2.1 DOC-7812562=	Provides information on all supported MIB[2]objects, support restrictions, traps, and alarms for the AXSM, PXM45, and RPM. PNNI is also supported.
Cisco MGX and SES PNNI Network Planning Guide DOC-7813543=	Provides guidelines for planning a PNNI network that uses the MGX 8850 switch and the BPX 8600 switch. When connected to a PNNI network, each BPX 8600 series switch requires a Service Expansion Shelf (SES) for PNNI route processing.
Cisco MGX Route Processor Module Installation and Configuration Guide, Release 2.1 DOC-7812510=	Describes how to install and configure the MGX Route Processor Module (RPM-PR) in the MGX 8850 Release 2.1 switch. Also provides site preparation, troubleshooting, maintenance, cable and connector specifications, and basic IOS configuration information.

1. CLI = command line interface
2. MIB = Management Information Base

SES PNNI Release 1.1 Documentation

The product documentation that contains information for the understanding, the installation, and the operation of the Service Expansion Shelf (SES) PNNI Controller is listed in Table 4.

Table 4 SES PNNI Controller Release 1.1 Documentation

Title	Description
Cisco SES PNNI Controller Software Configuration Guide, Release 1.1 DOC-7813539=	Describes how to configure, operate, and maintain the SES PNNI Controller.
Cisco SES PNNI Controller Software Command Reference, Release 1.1 DOC-7813541=	Provides a description of the commands used to configure and operate the SES PNNI Controller.
Cisco MGX and SES PNNI Network Planning Guide DOC-7813543=	Provides guidelines for planning a PNNI network that uses the MGX 8850 switch and the BPX 8600 switch. When connected to a PNNI network, each BPX 8600 series switch requires a SES for PNNI route processing.

Cisco WAN Switching Software, Release 9.3 Documentation

The product documentation for the installation and operation of the Cisco WAN Switching Software Release 9.3 is listed in Table 5.

Table 5 Cisco WAN Switching Release 9.3 Documentation

Title	Description
Cisco BPX 8600 Series Installation and Configuration, Release 9.3.30 DOC-7812907=	Provides a general description and technical details of the BPX broadband switch.
Cisco WAN Switching Command Reference, Release 9.3.30 DOC-7812906=	Provides detailed information on the general command line interface commands.
Cisco IGX 8400 Series Installation Guide, Release 9.3.30 OL-1165-01 (online only)	Provides hardware installation and basic configuration information for IGX 8400 Series switches running Switch Software Release 9.3.30 or earlier.
Cisco IGX 8400 Series Provisioning Guide, Release 9.3.30 OL-1166-01 (online only)	Provides information for configuration and provisioning of selected services for the IGX 8400 Series switches running Switch Software Release 9.3.30 or earlier.
Cisco IGX 8400 Series Regulatory Compliance and Safety Information DOC-7813227=	Provides regulatory compliance, product warnings, and safety recommendations for the IGX 8400 Series switch.

MGX 8850 Multiservice Switch, Release 1.1.40 Documentation

The product documentation that contains information for the installation and operation of the MGX 8850 Multiservice Switch is listed in Table 6.

Table 6 MGX 8850 Multiservice Gateway Documentation

Title	Description
Cisco MGX 8850 Multiservice Switch Installation and Configuration, Release 1.1.3 DOC-7811223=	Provides installation instructions for the MGX 8850 multiservice switch.
Cisco MGX 8800 Series Switch Command Reference, Release 1.1.3. DOC-7811210=	Provides detailed information on the general command line for the MGX 8850 switch.
Cisco MGX 8800 Series Switch System Error Messages, Release 1.1.3 DOC-7811240=	Provides error message descriptions and recovery procedures.
Cisco MGX 8850 Multiservice Switch Overview, Release 1.1.3 OL-1154-01 (online only)	Provides a technical description of the system components and functionary of the MGX 8850 multiservice switch from a technical perspective.
Cisco MGX Route Processor Module Installation and Configuration Guide, Release 1.1 DOC-7812278=	Describes how to install and configure the MGX Route Processor Module (RPM/B and RPM-PR) in the MGX 8850, MGX 8250, and MGX 8230 Release 1 switch. Also provides site preparation, troubleshooting, maintenance, cable and connector specifications, and basic IOS configuration information.
1.1.40 Version Software Release Notes Cisco WAN MGX 8850, MGX 8230, and MGX 8250 Switches DOC-7813594=	Provides new feature, upgrade, and compatibility information, as well as known and resolved anomalies.

MGX 8250 Edge Concentrator, Release 1.1.40 Documentation

The documentation that contains information for the installation and operation of the MGX 8250 Edge Concentrator is listed in Table 7.

Table 7 MGX 8250 Multiservice Gateway Documentation

Title	Description
Cisco MGX 8250 Edge Concentrator Installation and Configuration, Release 1.1.3 DOC-7811217=	Provides installation instructions for the MGX 8250 Edge Concentrator.
Cisco MGX 8250 Multiservice Gateway Command Reference, Release 1.1.3 DOC-7811212=	Provides detailed information on the general command line interface commands.

Table 7 MGX 8250 Multiservice Gateway Documentation (continued)

Title	Description
Cisco MGX 8250 Multiservice Gateway Error Messages, Release 1.1.3 DOC-7811216=	Provides error message descriptions and recovery procedures.
Cisco MGX 8250 Edge Concentrator Overview, Release 1.1.3 DOC-7811576=	Describes the system components and functionality of the MGX 8250 edge concentrator from a technical perspective.
Cisco MGX Route Processor Module Installation and Configuration Guide, Release 1.1 DOC-7812278=	Describes how to install and configure the MGX Route Processor Module (RPM/B and RPM-PR) in the MGX 8850, MGX 8250, and MGX 8230 Release 1 switch. Also provides site preparation, troubleshooting, maintenance, cable and connector specifications, and basic IOS configuration information.
1.1.40 Version Software Release Notes Cisco WAN MGX 8850, MGX 8230, and MGX 8250 Switches DOC-7813594=	Provides new feature, upgrade, and compatibility information, as well as known and resolved anomalies.

MGX 8230 Multiservice Gateway, Release 1.1.40 Documentation

The documentation that contains information for the installation and operation of the MGX 8230 Edge Concentrator is listed in Table 8.

Table 8 MGX 8230 Multiservice Gateway Documentation

Title	Description
Cisco MGX 8230 Edge Concentrator Installation and Configuration, Release 1.1.3 DOC-7811215=	Provides installation instructions for the MGX 8230 Edge Concentrator.
Cisco MGX 8230 Multiservice Gateway Command Reference, Release 1.1.3 DOC-7811211=	Provides detailed information on the general command line interface commands.
Cisco MGX 8230 Multiservice Gateway Error Messages, Release 1.1.3 DOC-78112113=	Provides error message descriptions and recovery procedures.
Cisco MGX 8230 Edge Concentrator Overview, Release 1.1.3 DOC-7812899=	Provides a technical description of the system components and functionary of the MGX 8250 edge concentrator from a technical perspective.

Table 8 MGX 8230 Multiservice Gateway Documentation (continued)

Title	Description
Cisco MGX Route Processor Module Installation and Configuration Guide, Release 1.1 DOC-7812278=	Describes how to install and configure the MGX Route Processor Module (RPM/B and RPM-PR) in the MGX 8850, MGX 8250, and MGX 8230 Release 1 switch. Also provides site preparation, troubleshooting, maintenance, cable and connector specifications, and basic IOS configuration information.
1.1.40 Version Software Release Notes Cisco WAN MGX 8850, MGX 8230, and MGX 8250 Switches DOC-7813594=	Provides new feature, upgrade, and compatibility information, as well as known and resolved anomalies.

Conventions

This publication uses the following conventions to convey instructions and information.

Command descriptions use these conventions:

- Commands and keywords are in **boldface**.
- Arguments for which you supply values are in *italics*.

Examples use these conventions:

- Terminal sessions and information the system displays are in `screen` font.
- Information you enter is in **`boldface screen`** font.

Notes use the following conventions and symbols:

Note Means *reader take note*. Notes contain helpful suggestions or references to materials not contained in this manual.

Obtaining Documentation

The following sections explain how to obtain documentation from Cisco Systems.

World Wide Web

You can access the most current Cisco documentation on the World Wide Web at the following URL:

http://www.cisco.com

Translated documentation is available at the following URL:

http://www.cisco.com/public/countries_languages.shtml

Documentation CD-ROM

Cisco documentation and additional literature are available in a Cisco Documentation CD-ROM package, which is shipped with your product. The Documentation CD-ROM is updated monthly and may be more current than printed documentation. The CD-ROM package is available as a single unit or through an annual subscription.

Ordering Documentation

Cisco documentation is available in the following ways:

- Registered Cisco Direct Customers can order Cisco product documentation from the Networking Products MarketPlace:

  http://www.cisco.com/cgi-bin/order/order_root.pl

- Registered Cisco.com users can order the Documentation CD-ROM through the online Subscription Store:

  http://www.cisco.com/go/subscription

- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco corporate headquarters (California, USA) at 408 526-7208 or, elsewhere in North America, by calling 800 553-NETS (6387).

Documentation Feedback

If you are reading Cisco product documentation on Cisco.com, you can submit technical comments electronically. Click **Feedback** at the top of the Cisco Documentation home page. After you complete the form, print it out and fax it to Cisco at 408 527-0730.

You can e-mail your comments to bug-doc@cisco.com.

To submit your comments by mail, use the response card behind the front cover of your document, or write to the following address:

Cisco Systems
Attn: Document Resource Connection
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Obtaining Technical Assistance

Cisco provides Cisco.com as a starting point for all technical assistance. Customers and partners can obtain documentation, troubleshooting tips, and sample configurations from online tools by using the Cisco Technical Assistance Center (TAC) Web Site. Cisco.com registered users have complete access to the technical support resources on the Cisco TAC Web Site.

Cisco.com

Cisco.com is the foundation of a suite of interactive, networked services that provides immediate, open access to Cisco information, networking solutions, services, programs, and resources at any time, from anywhere in the world.

Cisco.com is a highly integrated Internet application and a powerful, easy-to-use tool that provides a broad range of features and services to help you to

- Streamline business processes and improve productivity
- Resolve technical issues with online support
- Download and test software packages
- Order Cisco learning materials and merchandise
- Register for online skill assessment, training, and certification programs

You can self-register on Cisco.com to obtain customized information and service. To access Cisco.com, go to the following URL:

http://www.cisco.com

Technical Assistance Center

The Cisco TAC is available to all customers who need technical assistance with a Cisco product, technology, or solution. Two types of support are available through the Cisco TAC: the Cisco TAC Web Site and the Cisco TAC Escalation Center.

Inquiries to Cisco TAC are categorized according to the urgency of the issue:

- Priority level 4 (P4)—You need information or assistance concerning Cisco product capabilities, product installation, or basic product configuration.
- Priority level 3 (P3)—Your network performance is degraded. Network functionality is noticeably impaired, but most business operations continue.
- Priority level 2 (P2)—Your production network is severely degraded, affecting significant aspects of business operations. No workaround is available.
- Priority level 1 (P1)—Your production network is down, and a critical impact to business operations will occur if service is not restored quickly. No workaround is available.

Which Cisco TAC resource you choose is based on the priority of the problem and the conditions of service contracts, when applicable.

Cisco TAC Web Site

The Cisco TAC Web Site allows you to resolve P3 and P4 issues yourself, saving both cost and time. The site provides around-the-clock access to online tools, knowledge bases, and software. To access the Cisco TAC Web Site, go to the following URL:

http://www.cisco.com/tac

All customers, partners, and resellers who have a valid Cisco services contract have complete access to the technical support resources on the Cisco TAC Web Site. The Cisco TAC Web Site requires a Cisco.com login ID and password. If you have a valid service contract but do not have a login ID or password, go to the following URL to register:

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

If you cannot resolve your technical issues by using the Cisco TAC Web Site, and you are a Cisco.com registered user, you can open a case online by using the TAC Case Open tool at the following URL:

http://www.cisco.com/tac/caseopen

If you have Internet access, it is recommended that you open P3 and P4 cases through the Cisco TAC Web Site.

Cisco TAC Escalation Center

The Cisco TAC Escalation Center addresses issues that are classified as priority level 1 or priority level 2; these classifications are assigned when severe network degradation significantly impacts business operations. When you contact the TAC Escalation Center with a P1 or P2 problem, a Cisco TAC engineer will automatically open a case.

To obtain a directory of toll-free Cisco TAC telephone numbers for your country, go to the following URL:

http://www.cisco.com/warp/public/687/Directory/DirTAC.shtml

Before calling, please check with your network operations center to determine the level of Cisco support services to which your company is entitled; for example, SMARTnet, SMARTnet Onsite, or Network Supported Accounts (NSA). In addition, please have available your service agreement number and your product serial number.

CHAPTER 1

MIBs Supported

This chapter describes the Management Information Bases (MIBs) supported by AXSM, AXSM-E, PXM45, PNNI, and RPM. Existing MIBs include those public MIBs that are part of the Internet Official Protocol Standards and are published as RFC text files. Another category of existing MIBs are Cisco Enterprise MIBs, which are common for Cisco products. In both categories, AXSM and PXM45 implementation do not support all of the tables defined in a particular MIB.

Contents of this chapter include:

- Standard MIBs
- Cisco Enterprise MIBs
- MIB Support Restrictions
- OID Assignments
- Cisco Products MIB

Standard MIBs

Not all of the objects in a MIB are supported by PXM45, AXSM, AXSM-E, or RPM. Table 1-1 lists the standard MIBs for the applicable service module.

Note Restrictions on the use of these MIBs are listed in Table 1-3 and Table 1-4.

Table 1-1 Standard MIBs

MIB	RFC Number	MIB File Name	Supported Objects	Service Module
IF MIB	RFC 2683	IF-MIB.my	`ifTable`, `ifXTable`, `ifMtu`	PXM45 AXSM AXSM-E RPM[1]
IANAifType MIB	RFC 1573	IANAifType-MIB.my	All objects	PXM45 AXSM
ATM Forum TC MIB	—	ATM-FORUM-TC-MIB.my	All objects	PXM45

Table 1-1 Standard MIBs (continued)

MIB	RFC Number	MIB File Name	Supported Objects	Service Module

ENTITY MIB	RFC 2 737	ENTITY-MIB.my	`entPhysicalTable`
			`entPhysicalContainsTable`
			`entLastChangeTime`
				PXM45

RS 232 MIB	RFC 1659	RS-232-MIB.my	`rs232Number`
			`rs232PortTable`
			`rs232AsyncPortTable`
				PXM45
				PXM1

SNMP v2 MIB	RFC 1907	SNMPv2-MIB.my	All objects	PXM45
SNMP v2 SMI	RFC 2578	SNMPv2-SMI.my	Object Identifier Definitions	PXM45 AXSM RPM[1]
SNMP v2 TC	RFC 2579	SNMPv2-TC.my	Textual Conventions used in MIBs	PXM45 AXSM RPM[1]
SNMP v2 CONF	RFC 2580	SNMPv2-CONF.my	Conformance Statements used in MIBs	PXM45 AXSM RPM[1]

SONET MIB	RFC 2558	SONET-MIB.my	`sonetMediumTable`[2]
			`sonetSectionCurrentTable`
			`sonetSectionIntervalTable`
			`sonetLineCurrentTable`
			`sonetLineIntervalTable`
			`sonetFarEndLineCurrentTable`
			`sonetPathCurrentTable1`
			`sonetPathIntervalTable`
			`sonetFarEndPathCurrentTable`
			`sonetFarEndPathIntervalTable`
				AXSM

DS3 MIB	RFC 2496	DS3-MIB.my	`dsx3CircuitIdentifier`
			`dsx3ConfigTable`[2]
			`dsx3CurrentTable`
			`dsx3IntervalTable`
			`dsx3InvalidIntervals`
			`dsx3LineStatusLastChange`
			`dsx3LoopbackStatus`
			`dsx3TotalTable`
			`dsx3ValidIntervals`
				AXSM

Table 1-1 Standard MIBs (continued)

MIB	RFC Number	MIB File Name	Supported Objects	Service Module

ATM PNNI MIB	ATM Forum PNNI MIB	PNNI-MIB.my	`pnniBase Group`
			`pnniNodeTable`[3]
			`pnniNodePglTable`
			`pnniSummaryAddressTable`
			`pnniLinkTable`
			`pnniNodetimerTable`
			`pnniNodeSvccTable`
			`pnniNodeSscopeMappingTable`
			Conformance and Compliance Statements Used in MIBs
				PXM45

ATM MIB	RFC 2515	ATM-MIB.my	`atmInterfaceConfTable`[4]	PXM45

1. This MIB is implemented in PXM45.
2. Usage is limited.
3. pnniNodeDomain Name is not supported.
4. This object supports only atmInterfaceMyNeighborIpAddress and atmInterface MyNeighborIfName as read-only.

Cisco Enterprise MIBs

Table 1-2 lists the Cisco Enterprise MIBs supported by PXM45, AXSM, AXSM-E, and RPM.

The CISCO-WAN-SVC-MIB.my file is defined for Switched Virtual Circuit (SVC) and PNNI management.

Note Not all of the objects in a MIB are supported by PXM45 and AXSM.

Table 1-2 Cisco Enterprise MIBs

MIB	MIB File Name	Supported Objects	Service Module
CISCO FRU CONTROL MIB	CISCO-ENTITY-FRU-CONTROL-MIB.my	`cefcModuleTable`	PXM45
SENSOR MIB	CISCO-ENTITY-SENSOR-MIB.my	All objects	PXM45
CISCO PRODUCTS MIB	CISCO-PRODUCTS-MIB.my	Only OIDs[1] used for `sysObjectID` object for MSSBU products	PXM45
CISCO SMI	CISCO-SMI.my	CISCO-Specific Structure of Management Information	PXM45 RPM[2]
CISCO WAN SCM MIB[3]	CISCO-WAN-SCM-MIB.my	All objects	PXM45

Table 1-2 Cisco Enterprise MIBs (continued)

MIB	MIB File Name	Supported Objects	Service Module
ATM Connection Statistics MIB[3]	CISCO-WAN-ATM-CONN-STAT-MIB.my	cwacsIngRcvCLP0 cwacsIngRcvCLP1 cwacsIngCLP0UpcDiscard cwacsIngCLP1UpcDiscard cwacsIngCLP0UpcTagged cwacsIngRcvEFCI1 cwacsIngRcvEOF1 cwacsIngVCQueueDepth cwacsEgrRcvCLP0 cwacsEgrRcvCLP1 cwacsEgrRcvEFCI1 cwacsEgrRcvEOF1 cwacsEgrVCQueueDepth cwacsStatsClear	AXSM-E
Service Class Template MIB	CISCO-WAN-SCT-MIB.my	All objects	AXSM AXSM-E
Cisco WAN ATM Class of Service Buffer MIB[3]	CISCO-WAN-ATM-COSB-MIB.my	All objects	AXSM-E
CISCO BULK FILE MIB	CISCO-BULK-FILE-MIB.my	cbfDefineFileTable cbfDefineFileEntry cbfDefineFileIndex cbfDefineFileName cbfDefineFileStorage cbfDefineFileFormat cbfDefineFileNow cbfDefineFileEntryStatus cbfDefineObjectTable cbfDefineObjectEntry cbfDefineObjectIndex cbfStatusFileTable cbfStatusFileEntry cbfStatusFileIndex cbfStatusFileState cbfStatusFileCompletionTime cbfStatusFileEntryStatus Conformance and Compliance Statements Used in MIBs	AXSM PXM45 RPM[2]
CISCO TC	CISCO-TC.my	CISCO-Specific Textual Conventions	PXM45 RPM[2]
CISCO SYSTEM MIB	CISCO-SYSTEM-MIB.my	All objects	PXM45
ENTITY VENDORTYPE MIB	CISCO-ENTITY-VENDORTYPE-MIB.my	Those OIDs used for entphysicalVendorType Object for StrataCom products	PXM45
CISCO VSI CONTROLLER	CISCO-VSI-CONTROLLER-MIB.my	All objects	PXM45
CISCO SONET MIB	CISCO-SONET-MIB.my	CISCO-Specific extensions to SONET-MIB	AXSM AXSM-E

Table 1-2 Cisco Enterprise MIBs (continued)

MIB	MIB File Name	Supported Objects	Service Module
CISCO WAN SONET MIB[3]	CISCO-WAN-SONET-MIB.my	SONET Section Alarm Table Line Alarm Table Path Alarm Table	AXSM
RESOURCE PARTITION MIB	CISCO-WAN-RSRC-PART-MIB.my	Resource Partition MIB for the ATM Virtual Interfaces	AXSM RPM[2]
CISCO DS3 MIB	CISCO-DS3-MIB.my	Cisco-Specific Extensions to DS3-MIB cds3TraceAlarm cds3RcvAISCount cds3LCVPrevious24Hr cds3LESPrevious24Hr cds3PCVPrevious24Hr cds3PESPrevious24Hr cds3PSESPrevious24Hr cds3SEFSPrevious24Hr cds3UASPrevious24Hr cds3CCVPrevious24Hr cds3CESPrevious24Hr cds3CSESPrevious24Hr	AXSM AXSM-E
ATM VIRTUAL IF MIB	CISCO-ATM-VIRTUAL-IF-MIB.my	All objects	AXSM AXSM-E
ATM PHYSICAL LAYER MIB	CISCO-ATM-CELL-LAYER-MIB.my	All objects	AXSM AXSM-E
ATM Connections MIB[3]	CISCO-WAN-ATM-CONN-MIB.my	All objects	AXSM AXSM-E RPM[2]
CISCO WAN Feeder MIB[3]	CISCO-WAN-FEEDER-MIB.my	All objects	AXSM
CISCO WAN Module MIB	CISCO-WAN-MODULE-MIB.my	All objects	AXSM AXSM-E
CISCO SVC[3]	CISCO-WAN-SVC-MIB.my	ciscoWANSvcInfo Group cwspConfigTable cwspCacConfigTable cwspOperationTable cwspLoadTable cwspAddressTable cwspCallStatsTable cwspSigStatsTable cwspConnTraceGroup cwspConnTraceCntlTable cwspConnTraceTable	PXM45
RPM Sub-Interface MIB[3]	CISCO-WAN-RPM-SUBIF-MIB.my	All objects	RPM[2]
RPM Connection Extension MIB[3]	CISCO-WAN-RPM-CONN-EXT-MIB.my	All objects	RPM[2]
RTM MIB[3]	RTM-MIB.my	All objects	PXM45
ERROR STATUS MIB[3]	ERR-STATUS-MIB.my	All objects	PXM45 RPM[2]

Table 1-2 Cisco Enterprise MIBs (continued)

MIB	MIB File Name	Supported Objects	Service Module
BASIS SHELF MIB[3]: For Shelf and Redundancy Table	BASIS-SHELF-MIB.my	statsMasterIpAddress statsCollectionInterval statsBucketInterval userName shelfIntegratedAlarm shelfNum redPrimarySlotNum redRowStatus redPrimaryState redSecondarySlotNum redSecondaryState redType redCoveringSlot	PXM45
MODULE MIB for TRAPS[3]	BASIS-GENERIC-MIB.my	moduleTrapAlarmSeverity	PXM45
GENERIC MIB for TRAPS[3]	GENERICOBJECT-MIB.my	genericTimeStamp	PXM45

1. OID = object identifier
2. This MIB is implemented in PXM45.
3. This MIB is also defined under the StrataCom Enterprise.

MIB Support Restrictions

For certain standard MIB objects, the access defined in the MIB and the access as implemented AXSM, AXSM-E, and PXM45 are different, with a different default value in certain cases. These constitute MIB support limitations.

PXM Support Restrictions

Table 1-3 lists the MIB supported limitation objects. In all cases, the access defined by the MIB is for read-write, while only read-only is supported in the PXM45.

Table 1-3 MIB Support Limitations for PXM

MIB	MIB File Name	Unsupported Table/Objects
SNMPv2 MIB	SNMPv2-MIB.my	sysORLastChange sysORTable
IF MIB	IF-MIB.my	ifInOctets ifInUcastPkts ifInNUcastPkts ifInDiscards ifInErrors ifInUnknownProtos ifOutOctets ifOutUcastPkts ifOutNUcastPkts ifOutDiscards ifOutErrors ifInMulticastPkts ifOutQLen ifSpecific ifInMulticastPkts ifInBroadcastPkts ifOutMulticastPkts ifOutBroadcastPkts ifHCInOctets ifHCInUcastPkts ifHCInMulticastPkts ifHCInBroadcastPkts ifHCOutOctets ifHCOutUcastPkts ifHCOutMulticastPkts ifHCOutBroadcastPkts ifTableLastChange ifStackTable ifRcvAddressTable linkDown TRAP linkUp Trap All the deprecated Objects
ENTITY MIB	ENTITY-MIB.my	entPhysicalAssetID entPhysicalAlias entLogicalTable entLPMappingTable entAliasMappingTable entConfigChange TRAP
CISCO ENTITY FRU CONTROL MIB	CISCO-ENTITY-FRU-CONTROL-MIB.my	cefcFRUPowerSupplyGroupTable

Table 1-3 MIB Support Limitations for PXM (continued)

MIB	MIB File Name	Unsupported Table/Objects
BASIS SHELF MIB	BASIS-SHELF-MIB.my	redPrimaryType redSecondaryType redFeature redLineModuleType shelfTable shelfNumValidEntries shelfDate shelfTime shelfTmZn shelfTmZnGMTOff shelfBkplnType shelfBkplnSerialNum shelfAlarmCardBitMap apsIpAddress redundantApsIpAddress axisFeederTkNo axisSvcBillingColInterval axisSvcBillingBucketInterval axisSvcBilling
BASIS GENERIC MIB	BASIS-GENERIC-MIB.my	All objects except moduleTrapAlarmSeverity
GENERICOBJECT MIB	GENERICOBJECT-MIB.my	genericLineNum genericLineType

AXSM Support Restrictions

AXSM objects with MIB support limitations are listed in Table 1-4. In all cases, the access defined by the MIB is for read-write, while AXSM implementation supports read-only. The MIB file can be identified by the first part of the object name, SONET refers to the file SONET-MIB.my, and dsx3 refers to the DS3-MIB.my file.

Table 1-4 MIB Support Limitations for AXSM

Object Name	Access Defined in MIB	Access as Implemented in AXSM	Default Value/Equivalent Object
sonetPathCurrentWidth	read-write	read-only	csConfigType
sonetMediumType	read-write	read-only	none
sonetMediumLineCoding	read-write	read-only	none
sonetMediumLineType	read-write	read-only	none
sonetMediumCircuitIdentifier	read-write	read-only	none
sonetMediumLoopbackConfig	read-write	read-only	not supported
dsx3LineType	read-write	read-only	other (1)
dsx3LineCoding	read-write	read-only	none
dsx3LineStatusChangeTrapEnable	read-write	read-only	disabled (1)
dsx3Channelization	read-write	read-only	disabled (1)

Unsupported MIB Objects

The tables or objects that are unsupported by AXSM for the standard MIBs are listed in Table 1-5.

Table 1-5 Standard Unsupported MIB Objects for AXSM

MIB File	Unsupported Table or Objects
IF-MIB.my	ifInNUcastPkts
	ifInOctets
	ifInUcastPkts
	ifInDiscards
	ifInErrors
	ifInUnknownProtos
	ifOutOctets
	ifOutUcastPkts
	ifOutNUcastPkts
	ifOutDiscards
	ifOutErrors
	ifInMulticastPkts
	ifOutQlen
	ifSpecific
	ifInMulticastPkts
	ifInBroadcastPkts
	ifOutMulticastPkts
	ifOutBroadcastPkts
	ifHCInOctets
	ifHCInUcastPkts
	ifHCInMulticastPkts
	ifHCInBroadcastPkts
	ifHCOutOctets
	ifHCOutUcastPkts
	ifHCOutMulticastPkts
	ifHCOutBroadcastPkts
	ifTableLastChange
	ifStackTable
	ifRcvAddressTable
	linkDownTRAP
	linkUpTrap

Table 1-5 Standard Unsupported MIB Objects for AXSM (continued)

MIB File	Unsupported Table or Objects
SONET-MIB.my	sonetMediumLoopbackConfig
	sonetVTCurrentTable
	sonetVTIntervalTable
	sonetFarEndVTCurrentTable
	sonetFarEndVTIntervalTable
DS3-MIB.my	dsx3FarEndConfigTable
	dsx3FracTable
	dsx3FarEndCurrentTable
	dsx3FarEndIntervalTable
	dsx3FarEndTotalTable
	dsx3LineStatusChange[1]
	dsx3TimeElasped

1. Denotes a Trap.

The unsupported objects used for the Cisco Enterprise MIB for AXSM are listed in Table 1-6.

Table 1-6 Cisco Enterprise MIB Unsupported Objects for AXSM

MIB File	Unsupported Object
CISCO-DS3-MIB.my	cds3CPECount
CISCO-WAN-ATM-CONN-STAT-MIB.my	cwacsIngXmtCLP0
	cwacsIngXmtCLP1
	cwacsIngCLP0CoSDiscard
	cwacsIngCLP1CoSDiscard
	cwacsIngRcvEFCI0
	cwacsIngEFCI0Discard
	cwacsIngEFCI1Discard
	cwacsIngRcvOAM
	cwacsIngOAMDiscard
	cwacsIngRcvRM
	cwacsIngRMDiscard
	cwacsIngXmtFRm
	cwacsIngXmtBRmFsRm
	cwacsIngEOF1Discard
	cwacsIngACR
	cwacsEgrXmtCLP0
	cwacsEgrXmtCLP1
	cwacsEgrCLP0CoSDiscard
	cwacsEgrCLP1CoSDiscard

Table 1-6 Cisco Enterprise MIB Unsupported Objects for AXSM (continued)

MIB File	Unsupported Object
CISCO-WAN-ATM-CONN-STAT-MIB.my (continued)	cwacsEgrRcvEFCI0
	cwacsEgrEFCI0Discard
	cwacsEgrEFCI1Discard
	cwacsEgrRcvOAM
	cwacsEgrOAMDiscard
	cwacsEgrRcvRM
	cwacsEgrRMDiscard
	cwacsEgrXmtFRm
	cwacsEgrXmtBRmFsRm
	cwacsEgrEOF1Discard
	cwacsEgrACR

Interface Terminologies

The ifTable defined in IF-MIB, which is supported by AXSM, PXM45, and RPM, has entries for the various physical and logical interfaces. The terminologies used for the various interfaces are listed in Table 1-7. These interfaces are identified by the ifType MIB object in ifTable.

Table 1-7 Interface Terminologies Used

Interface	Identification	ifType Values
physical interface	Identifies the interface for physical lines.	ds3(30), sonet (39), ds1 (18), ethernetCsmacd(6)
ATM physical interface (ATM Phy Interface)	Identifies the interface for the ATM cell layer on a physical interface.	atm(37)
ATM virtual interface	Identifies the logical interface configured on a physical interface.	atmVirtual(14)

OID Assignments

This MIB contains OID assignments for Cisco components for use by the ENTITY-MIB for PXM. This MIB module defines the object identifiers that are assigned to various components on Cisco products, which are used by the entPhysicalTable of the ENTITY-MIB to uniquely identify the type of each physical entry.

Table 1-8 lists the values for entPhysicalVendorType object in the switch.

Chassis, Backplane, Slots, and Fans

Table 1-8 lists the OID assignments for chassis, backplane, slots, and fans.

Table 1-8 OID Assignments—Chassis, Backplane, Slots, and Fans

Component Name	OID Name	OID Value
MGX 8850 chassis	cevChassisMGX8850	cevChasis 86
BPX SES chassis	cevchassis	cevChassis 101
MGX 8850 backplane	cevBackplaneMGX8850	cevBackplane 9
BPX-SES backplane	cevBackplaneBPXSES	cevBackplane 11
Processor Switch Module (PXM) slot	cevContainerPsmSlot	cevContainer 31
Service Module (SM) slot	cevContainerSmSlot	cevContainer 32
Service Redundancy Module (SRM) slot	cevContainerSrmSlot	cevContainer 34
MGX 8800 fan tray (9 fans)	cevContainerMGX8800FanTray	cevContainer 33
SES fan tray (8 fans)	sevContainerSESFanTray	cevContainer 36
MGX 8800 AC power supply tray	cevContainerMGX8800ACPSTray	cevContainer 35
MGX 8800 single pulse fan	cevFanSinglePulse	cevFan 4
MGX 8800 dual pulse fan	cevFanDualPulse	cevFan 5
MGX 8800 power entry module (PEM)	cevPowerSupplyMGX8800Pem	cevPowerSupply 17
MGX 8800 temperature sensor	cevSensorMGX8800Temp	cevSensor 11
MGX 8800 power supply voltage level monitor (DC Level Monitor)	cevSensorMGX8800PSVoltage	cevSensor 12
Fan speed sensor for MGX 8850, SES	cevSensorMGX8800FanSpeed	cevSensor 13
Alternating current (AC) power supply, 1200 Watts	cevPowerSupplyAC1200W	cevPowerSupply 10
Alternating current (AC) power supply, 1050 Watts	cevPowerSupplyAC1050W	cevPowerSupply 23

Front Modules

Table 1-9 lists the OID assignments for Front modules as defined in CISCO-ENTITY-VENDORTYPE-OID-MIB.

Table 1-9 OID Assignments—Front Modules

Component Name	OID Name	OID Value
PXM1 module	cevCpuPSM1Gps	cevModuleCpuType 12
PXM1-2-T3E3	cevCpuPsm12t3e3	cevModuleCpuType 16
PXM1-4-155	cevCpuPsm14oc3	cevModuleCpuType 17
PXM1-622	cevCpuPsm11oc12	cevModuleCpuType 18
PXM45	cevCpuPSM45Gbps	cevModuleCpuType 13
AXSM-1-2488	cevAxsm1Oc48	cevModuleWASCommonCards 9
AXSM-4-622	cevAxsm4Oc12	cevModuleWASCommonCards 10
AXSM-2-622	cevAxsm2oc12	cevModuleWASCommonCards 32
AXSM-8-155	cevAxsm8Oc3	cevModuleWASCommonCards 11
AXSM-16-155	cevAxsm16Oc3	cevModuleWASCommonCards 12
AXSM-16-T3E3	cevAxsm16T3E3	cevModuleWASCommonCards 13
AXSM-1-2488/B	cevAxsm1Oc48B	cevModuleWASCommonCards 35
AXSM-4-622/B	cevAxsm4Oc12B	cevModuleWASCommonCards 36
AXSM-16-155/B	cevAxsm16Oc3B	cevModuleWASCommonCards 37
MGX8800 RPM	cevMGXRpm	cevModuleWASCommonCards 18
MGX RPM-PR	cevMGXRpmPr	cevModuleWASCommonCards 34
AXSM-2-622-E	cevEnhAxsm2oc12	cevModuleWASCommonCards 21
AXSM-8-155-E	cevEnhAxsm8oc3	cevModuleWASCommonCards 23
AXSM-16-T3E3-E	cevEnhAxsm16t3e3	cevModuleWASCommonCards 27

Line Modules

Table 1-10 lists the OID assignments for line modules.

Table 1-10 OID Assignments—Line Modules

Component Name	OID Name	OID Value
PXM-UI	cevLmPsmUI	cevModuleWASBackCards 11
PXM-HD	cevLmPsmHD	cevModuleWASBackCards 28
PXM-UI-S3	cevLmS3Ui	cevModuleWASBackCards 17
MGX-MMF-4-155	cevLmMmf4oc3	cevModuleWASBackCards 12
MGX-SMFIR-4-155	cevLmSmfIr4oc3	cevModuleWASBackCards 13
MGX-SMFLR-4-155	cevLmSmfLr4oc3	cevModuleWASBackCards 14

Table 1-10 OID Assignments—Line Modules (continued)

Component Name	OID Name	OID Value
MGX-SMFIR-1-622	cevLmSmfIr1oc12	cevModuleWASBackCards 15
MGX-SMFLR-1-622	cevLmSmfLrloc12	cevModuleWASBackCards 16
SMFIR-1-2488	cevLmSmfIr1oc48	cevModuleWASBackCards 26
SMFLR-1-2488	cevLmSmfLr1oc48	cevModuleWASBackCards 27
SMFSR-1-2488	cevLmSmfSrloc48	cevModuleWASBackCards 34
SMFXLR-1-2488	cevLmSmfXlrloc48	cevModuleWASBackCards 35
SMFIR-1-2488/B	cevLmSmfIrlOc48B	cevModuleWASBackCards 37
SMFLR-1-2488/B	cevLmSmfLrlOc48B	cevModuleWASBackCards 38
SMFSR-1-2488/B	cevLmSmfSr1Oc48B	cevModuleWASBackCards 39
SMFIR-2-622	cevLmSmfIr2oc12	cevModuleWASBackCards 32
SMFLR-2-622	cevLmSmfLr2oc12	cevModuleWASBackCards 33
SMFIR-2-622/B	cevLmSmfIr2Oc12B	cevModuleWASBackCards 40
SMFLR-2-622/B	cevLmSmfLr2Oc12B	cevModuleWASBackCards 41
SMFIR-1-622/C	cevLmSmfIr1Oc12C	cevModuleWASBackCards 42
SMFLR-1-622/C	cevLmSmfLr1Oc12C	cevModuleWASBackCards 43
MMF-4-155/C	cevLmMmf4Oc3C	cevModuleWASBackCards 44
SMFIR-4-155/C	cevLmSmfIr4Oc3C	cevModuleWASBackCards 45
SMFLR-4-155/C	cevLmSmfLr4Oc3C	cevModuleWASBackCards 46
MMF-8-155	cevLmMmf8oc3	cevModuleWASBackCards 23
SMFIR-8-155	cevLmSmfIr8oc3	cevModuleWASBackCards 24
SMFLR-8-155	cevLmSmfLr8oc3	cevModuleWASBackCards 25
MMF-8-155-MT/B	cevLmMmf8Oc3B	cevModuleWASBackCards 47
SMFIR-8-155-LC/B	cevLmSmfIr8Oc3B	cevModuleWASBackCards 48
SMFLR-8-155-LC/B	cevLmSmfLr8Oc3B	cevModuleWASBackCards 49
SMB-8-T3	cevLmSmb8t3	cevModuleWASBackCards 29
SMB-8-E3	cevLmSmb8e3	cevModuleWASBackCards 30
SMB-4-155	cevLmSmb4stm1	cevModuleWASBackCards 31

Cisco Products MIB

The Cisco Products MIB contains OID assignments for the SNMPv2-MIB. This module defines the object identifiers that are assigned to various Cisco products. The `sysObjectID` values are assigned to the MGX series.

Product Object Identifiers

Table 1-11 lists the OID assignment for the switch.

Table 1-11 Product Specific Object Identifiers

Product Name	OID Name	OID Value
MGX 8850	ciscoMGX8850	ciscoProducts 228

CHAPTER 2

Standard MIB Objects

This chapter describes the standard Management Information Base (MIB) objects that are common to AXSM, AXSM-E, PXM45, and RPM.

Contents of this chapter include:

- SONET/SDH MIB Objects
- DS3/E3 MIB Objects
- Internet Assigned Numbers Authority Interface Textual Convention
- Interfaces Group MIB Objects
- RS-232 MIB Objects

Note RS-232 objects are used only for PXM45 and PXM1.

SONET/SDH MIB Objects

This section describes the SONET/SDH (Synchronous Digital Hierarchy) MIB objects, which reside in the SONET-MIB.my file.

Note The SONET MIB is supported by AXSM.

The SONET/SDH MIB objects include:

- SONET/SDH Medium Table
- SONET /SDH Section Current Table
- SONET/SDH Section Interval Table
- SONET/SDH Line Current Table
- SONET/SDH Line Interval Table
- SONET/SDH Far End Line Current Table
- SONET/SDH Path Current Table
- SONET/SDH Path Interval Table
- SONET/SDH Far End Path Current Table
- SONET/SDH Far End Path Interval Table

SONET/SDH Medium Table

For some applications, the SONET/SDH interfaces are electrical interfaces, not optical interfaces. Table 2-1 gives configuration information for both optical SONET/SDH interfaces and electrical SONET/SDH interfaces.

Table 2-1 SONET/SDH Medium Table Object Identifiers

Name	Object Identifier
sonetMediumTable	::= { sonetMedium 1 }
sonetMediumEntry	::= { sonetMediumTable 1 }
sonetMediumType	::= { sonetMediumEntry 1 }
sonetMediumTimeElapsed	::= { sonetMediumEntry 2 }
sonetMediumValidIntervals	::= { sonetMediumEntry 3 }
sonetMediumLineCoding	::= { sonetMediumEntry 4
sonetMediumLineType	::= { sonetMediumEntry 5 }
sonetMediumCircuitIdentifier	::= { sonetMediumEntry 6 }
sonetMediumInvalidIntervals	::= { sonetMediumEntry 7 }
sonetMediumLoopbackConfig	::= { sonetMediumEntry 8 }
sonetSESthresholdSet	::= { sonetMedium 2 }

The MIB objects are listed in Table 2-2.

Table 2-2 SONET/SDH Medium Table MIB Objects

Name	Syntax	Description	Default Value
sonetMedium Table	`SEQUENCE OF SonetMediumEntry`	Describes the SONET/SDH Medium table. Max-Access: not-accessible Status: current	none
sonetMedium Entry	`SonetMediumEntry`	Provides an entry in the SONET/SDH Medium table. Max-Access: not-accessible Status: current The index is `ifIndex`.	none
sonetMediumType	`INTEGER{sonet(1),sdh(2)}`	Identifies whether a SONET or a SDH signal is used across this interface. Max-Access: read-write Status: current	none

Table 2-2 SONET/SDH Medium Table MIB Objects (continued)

Name	Syntax	Description	Default Value
sonetMedium TimeElapsed	`Integer32 (1..900)`	Determines the number of seconds, including partial seconds, elapsed since the beginning of the current measurement period. If an adjustment is made in the system's time-of-day clock, the current interval exceeds the maximum value. Therefore, the agent can return the maximum value. Max-Access: read-only Status: current	none
sonetMedium ValidIntervals	`Integer32 (0..96)`	Determines the number of previous 15-minute intervals for which the data is collected. A SONET/SDH interface must support at least *n* intervals. The minimum value of *n* is four. The default of *n* is 32. The maximum value of *n* is 96. For example, the value is *n* unless the measurement was restarted within the last (*n*`*15`) minutes. Then, the value is the number of the complete 15-minute intervals for which the agent has at least some data. For example, where the agent is a proxy, some intervals are unavailable. In this case, this interval is the maximum interval number for which data is available. Max-Access: read-only Status: current	none
sonetMediumLine Coding	`INTEGER {sonetMediumOther(1), sonetMediumB3ZS(2), sonetMediumCMI(3), sonetMediumNRZ(4), sonetMediumRZ(5) }`	Describes the line coding for this interface. The Binary 3-Zero Substitution (B3ZS) and Coded Mark Inversion (CMI) are used for electrical SONET/SDH signals (STS-1 and STS-3). The Non-Return to Zero (NRZ) and the Return to Zero are used for optical SONET/SDH signals. Max-Access: read-write Status: current	none

Table 2-2 SONET/SDH Medium Table MIB Objects (continued)

Name	Syntax	Description	Default Value
sonetMediumLine Type	`INTEGER {sonetOther(1), sonetShortSingleMode(2), sonetLongSingleMode(3), sonetMultiMode(4), sonetCoax(5), sonetUTP(6)}`	Describes the line type for this interface. The following are the line types: • Short and long range • Single mode fiber or Multi-mode fiber interfaces • Coax and UTP for electrical interfaces The value `sonetOther` is used when the Line Type is not one of the listed values. Max-Access: read-write Status: current	none
sonetMedium CircuitIdentifier	`DisplayString (SIZE (0..255))`	Contains the transmission vendor circuit identifier for troubleshooting. **Note** The circuit identifier, if available, is also represented by `ifPhysAddress`. Max-Access: read-write Status: current	none
sonetMedium InvalidIntervals	`Integer32 (0..96)`	Determines the number of intervals in the range from 0 to `sonetMediumValidIntervals` for which no data is available. This object is 0 except in cases where the data for some intervals are not available, for example, in proxy situations. Max-Access: read-only Status: current	none

Table 2-2 SONET/SDH Medium Table MIB Objects (continued)

Name	Syntax	Description	Default Value
sonetMediumLoopbackConfig	`BITS {sonetNoLoop(0), sonetFacilityLoop(1), sonetTerminalLoop(2), sonetOtherLoop(3)}`	Describes the current loopback state of the SONET/SDH interface. The values include: • `sonetNoLoop`—Means not in the loopback state. A device that is not capable of performing a loopback on this interface can always return this value. • `sonetFacilityLoop`—Ensures that the received signal at this interface is looped back out through the corresponding transmitter in the return direction. • `sonetTerminalLoop`—Ensures that the signal about to be transmitted is connected to the associated incoming receiver. • `sonetOtherLoop`—Defines the loopbacks. Max-Access: read-write Status: current	none
sonetSESthresholdSet	`INTEGER {other(1), bellcore1991(2), ansi1993(3), itu1995(4), ansi1997(5)}`	Specifies an enumerated integer that indicates the recognized set of Severely Errored Seconds (SES) thresholds, which the agent uses for determining severely errored seconds and unavailable time. The values include: • `other(1)`—None of the following. • `bellcore1991(2)`—Bellcore TR-NWT-000253, 1991 [32], or ANSI T1M1.3/93-005R2, 1993 [22]. • `ansi1993(3)`—ANSI T1.231, 1993 [31], or Bellcore GR-253-CORE, Issue 2, 1995 [34]. • `itu1995(4)`—ITU Recommendation G.826, 1995 [33]. • `ansi1997(5)`—ANSI T1.231, 1997 [35]. If a manager changes the value of this object, the SES statistics collected before the change must be invalidated. Max-Access: read-write Status: current	none

SONET /SDH Section Current Table

The SONET/SDH Section Current Table contains various statistics that are collected for the current 15-minute interval.

The object identifiers are listed in Table 2-3.

Table 2-3 SONET/SDH Section Current Table Object Identifiers

Name	Object Identifier
sonetSectionCurrentTable	::= { sonetSection 1 }
sonetSectionCurrentEntry	::= { sonetSectionCurrentTable 1 }
sonetSectionCurrentStatus	::= { sonetSectionCurrentEntry 1 }
sonetSectionCurrentESs	::= { sonetSectionCurrentEntry 2 }
sonetSectionCurrentSESs	::= { sonetSectionCurrentEntry 3 }
sonetSectionCurrentSEFSs	::= { sonetSectionCurrentEntry 4 }
sonetSectionCurrentCVs	::= { sonetSectionCurrentEntry 5 }

The MIB objects are listed in Table 2-4.

Table 2-4 SONET/SDH Section Current Table MIB Objects

<table>
<tr><th>Name</th><th>Syntax</th><th>Description</th><th>Default Value</th></tr>
<tr><td>sonetSectionCurrentTable</td><td>SEQUENCE of SonetSectionCurrentEntry</td><td>Describes the SONET/SDH section current table.
Max-Access: not-accessible
Status: current</td><td>none</td></tr>
<tr><td>sonetSectionCurrentEntry</td><td>SonetSectionCurrentEntry</td><td>Provides an entry in the SONET/SDH section current table.
Max-Access: not-accessible
Status: current
The index is ifIndex.</td><td>none</td></tr>
<tr><td>sonetSectionCurrentStatus</td><td>Integer32 (1..6)</td><td>Indicates the status of the interface. The sonetSectionCurrentStatus is a bitmap represented as a sum; therefore, it can represent multiple defects simultaneously.
If no other flag is set, the sonetSectionNoDefect is set.
Max-Access: read-only
Status: current
The bit positions include:
<table>
<tr><th>Bit Position</th><th>Variable</th></tr>
<tr><td>1</td><td>sonetSectionNoDefect</td></tr>
<tr><td>2</td><td>sonetSectionLOS</td></tr>
<tr><td>4</td><td>sonetSectionLOF</td></tr>
</table></td><td>none</td></tr>
</table>

Table 2-4 SONET/SDH Section Current Table MIB Objects (continued)

Name	Syntax	Description	Default Value
sonetSectionCurrentESs	PerfCurrentCount	Specifies the counter associated with the number of errored seconds encountered by a SONET/SDH section in the current 15-minute interval. Max-Access: read-only Status: current	none
sonetSectionCurrentSESs	PerfCurrentCount	Specifies the counter associated with the number of severely errored seconds encountered by a SONET/SDH section in the current 15-minute interval. Max-Access: read-only Status: current	none
sonetSectionCurrentSEFSs	PerfCurrentCount	Specifies the counter associated with the number of severely errored framing seconds encountered by a SONET/SDH Section in the current 15-minute interval. Max-Access: read-only Status: current	none
sonetSectionCurrentCVs	PerfCurrentCount	Specifies the counter associated with the number of coding violations encountered by a SONET/SDH section in the current 15-minute interval. Max-Access: read-only Status: current	none

SONET/SDH Section Interval Table

The SONET/SDH Section Interval Table contains statistics that are collected by each system over the previous 24 hours. The past 24 hours is broken into 96 completed 15-minute intervals.

Each system is required to store at least four completed 15-minute intervals. The default value is 32 intervals.

The object identifiers are listed in Table 2-5.

Table 2-5 SONET/SDH Section Interval Table Object Identifiers

Name	Object Identifier
sonetSectionIntervalTable	::= { sonetSection 2 }
sonetSectionIntervalEntry	::= { sonetSectionIntervalTable 1 }
sonetSectionIntervalNumber	::= { sonetSectionIntervalEntry 1 }
sonetSectionIntervalESs	::= { sonetSectionIntervalEntry 2 }
sonetSectionIntervalSESs	::= { sonetSectionIntervalEntry 3 }
sonetSectionIntervalSEFSs	::= { sonetSectionIntervalEntry 4 }

Table 2-5 SONET/SDH Section Interval Table Object Identifiers (continued)

Name	Object Identifier
sonetSectionIntervalCVs	::= { sonetSectionIntervalEntry 5 }
sonetSectionIntervalValidData	::= { sonetSectionIntervalEntry 6 }

The MIB objects are listed in Table 2-6.

Table 2-6 SONET/SDH Section Interval Table MIB Objects

Name	Syntax	Description	Default Value
sonetSectionIntervalTable	`SEQUENCE OF SonetSectionIntervalEntry`	Describes the SONET/SDH section interval table. Max-Access: not-accessible Status: current	none
sonetSectionIntervalEntry	`SonetSectionIntervalEntry`	Provides an entry in the SONET/SDH section interval table. Max-Access: not-accessible Status: current The indexes include: • `ifIndex` • `sonetSectionIntervalNumber`	none
sonetSectionInterval Number	`Integer32 (1..96)`	Determines a number between 1 to 96, which identifies the interval for which the set of statistics is available. The interval identified by 1 is the most recently completed 15 minute interval, and the interval identified by *N* is the interval immediately preceding the one identified by *N*-1. Max-Access: not-accessible Status: current	none
sonetSectionIntervalESs	`PerfIntervalCount`	Specifies the counter associated with the number of Errored Seconds encountered by a SONET/SDH section interval table in a particular 15-minute interval in the past 24 hours. Max-Access: read-only Status: current	none
sonetSectionIntervalSESs	`PerfIntervalCount`	Specifies the counter associated with the number of severely Errored Seconds encountered by a SONET/SDH section interval table in a particular 15-minute interval in the past 24 hours. Max-Access: read-only Status: current	none

Table 2-6 SONET/SDH Section Interval Table MIB Objects (continued)

Name	Syntax	Description	Default Value
sonetSectionIntervalSEFSs	PerfIntervalCount	Specifies the counter associated with the number of severely Errored Framing Seconds encountered by a SONET/SDH section interval table in a particular 15-minute interval in the past 24 hours. Max-Access: read-only Status: current	none
sonetSectionIntervalCVs	PerfIntervalCount	Specifies the counter associated with the number of coding violations encountered by a SONET/SDH section interval table in a particular 15-minute interval in the past 24 hours. Max-Access: read-only Status: current	none
sonetSectionIntervalValid Data	TruthValue	Indicates if the data for this interval is valid. Max-Access: read-only Status: current	none

SONET/SDH Line Current Table

The SONET/SDH line current table contains statistics that are collected for the current 15-minute interval.

The object identifiers are listed in Table 2-7.

Table 2-7 SONET/SDH Line Current Table Object Identifiers

Name	Object Identifier
sonetLineCurrentTable	::= { sonetLine 1 }
sonetLineCurrentEntry	::= { sonetLineCurrentTable 1 }
sonetLineCurrentStatus	::= { sonetLineCurrentEntry 1 }
sonetLineCurrentESs	::= { sonetLineCurrentEntry 2 }
sonetLineCurrentSESs	::= { sonetLineCurrentEntry 3 }
sonetLineCurrentCVs	::= { sonetLineCurrentEntry 4 }
sonetLineCurrentUASs	::= { sonetLineCurrentEntry 5 }

The MIB objects are listed in Table 2-8.

Table 2-8 SONET/SDH Line Current Table MIB Objects

<table>
<tr><th>Name</th><th>Syntax</th><th>Description</th><th>Default Value</th></tr>
<tr><td>sonetLineCurrentTable</td><td>SEQUENCE OF SonetLineCurrentEntry</td><td>Describes the SONET/SDH line current table.
Max-Access: not-accessible
Status: current</td><td>none</td></tr>
<tr><td>sonetLineCurrentEntry</td><td>SonetLineCurrentEntry</td><td>Provides an entry in the SONET/SDH line current table.
Max-Access: not-accessible
Status: current
The index is ifIndex.</td><td>none</td></tr>
<tr><td>sonetLineCurrentStatus</td><td>Integer32 (1..6)</td><td>Indicates the status of the interface. The sonetLineCurrentStatus is a bit map represented as a sum; therefore, it can represent multiple defects simultaneously.
If no other flag is set, the sonetLineNoDefect is set.
Max-Access: read-only
Status: current
The bit positions include:
<table>
<tr><th>Bit Position</th><th>Variable</th></tr>
<tr><td>1</td><td>sonetLineNoDefect</td></tr>
<tr><td>2</td><td>sonetLineAIS</td></tr>
<tr><td>4</td><td>sonetLineRDI</td></tr>
</table></td><td>none</td></tr>
<tr><td>sonetLineCurrentESs</td><td>PerfCurrentCount</td><td>Specifies the counter associated with the number of Errored Seconds encountered by a SONET/SDH line in the current 15-minute interval.
Max-Access: read-only
Status: current</td><td>none</td></tr>
<tr><td>sonetLineCurrentSESs</td><td>PerfCurrentCount</td><td>Specifies the counter associated with the number of severely Errored Seconds encountered by a SONET/SDH line in the current 15-minute interval.
Max-Access: read-only
Status: current</td><td>none</td></tr>
</table>

Table 2-8 SONET/SDH Line Current Table MIB Objects (continued)

Name	Syntax	Description	Default Value
sonetLineCurrentCVs	PerfCurrentCount	Specifies the counter associated with the number of coding violations encountered by a SONET/SDH line in the current 15-minute interval. Max-Access: read-only Status: current	none
sonetLineCurrentUASs	PerfCurrentCount	Specifies the counter associated with the number of unavailable seconds encountered by a SONET/SDH line in the current 15-minute interval. Max-Access: read-only Status: current	none

SONET/SDH Line Interval Table

The SONET/SDH Line Interval Table contains statistics that are collected by each system for the previous 24 hours. The past 24 hours is broken into 96 completed 15-minute intervals. Each system is required to store at least four completed 15-minute intervals. The default is 32 intervals.

The object identifiers are listed in Table 2-9.

Table 2-9 SONET/SDH Line Interval Table Object Identifiers

Name	Object Identifier
sonetLineIntervalTable	::= { sonetLine 2 }
sonetLineIntervalEntry	::= { sonetLineIntervalTable 1 }
sonetLineIntervalNumber	::= { sonetLineIntervalEntry 1 }
sonetLineIntervalESs	::= { sonetLineIntervalEntry 2]
sonetLineIntervalSESs	::= { sonetLineIntervalEntry 3
sonetLineIntervalCVs	::= { sonetLineIntervalEntry 4 }
sonetLineIntervalUASs	::= { sonetLineIntervalEntry 5 }
sonetLineIntervalValidData	::= { sonetLineIntervalEntry 6 }

The MIB objects are listed in Table 2-10.

Table 2-10 SONET/SDH Line Interval Table MIB Objects

Name	Syntax	Description	Default Value
sonetLineIntervalTable	`SEQUENCE OF SonetLineIntervalEntry`	Describes the SONET/SDH line interval table. Max-Access: not-accessible Status: current	none
sonetLineIntervalEntry	`SonetLineIntervalEntry`	Provides an entry in the SONET/SDH line interval table. Max-Access: not-accessible Status: current The indexes include: • `ifIndex` • `sonetLineIntervalNumber`	none
sonetLineInterval Number	`Integer32 (1..96)`	Determines a number from 1 to 96, which identifies the interval for which the set of statistics is available. The interval identified by 1 is the most recently completed 15-minute interval, and the interval identified by *N* is the interval immediately preceding the one identified by *N*-1. Max-Access: not-accessible Status: current	none
sonetLineIntervalESs	`PerfIntervalCount`	Specifies the counter associated with the number of Errored Seconds encountered by a SONET/SDH line in a particular 15-minute interval in the past 24 hours. Max-Access: read-only Status: current	none
sonetLineIntervalSESs	`PerfIntervalCount`	Specifies the counter associated with the number of severely Errored Seconds encountered by a SONET/SDH line in a particular 15-minute interval in the past 24 hours. Max-Access: read-only Status: current	none
sonetLineIntervalCVs	`PerfIntervalCount`	Specifies the counter associated with the number of coding violations encountered by a SONET/SDH line in a particular 15-minute interval in the past 24 hours. Max-Access: read-only Status: current	none

Table 2-10 SONET/SDH Line Interval Table MIB Objects (continued)

Name	Syntax	Description	Default Value
sonetLineIntervalUASs	`PerfIntervalCount`	Specifies the counter associated with the number of unavailable seconds encountered by a SONET/SDH line in a particular 15-minute interval in the past 24 hours. Max-Access: read-only Status: current	none
sonetLineIntervalValid Data	`TruthValue`	Indicates if the data for this interval is valid. Max-Access: read-only Status: current	none

SONET/SDH Far End Line Current Table

The SONET/SDH far end line current table contains various statistics that are collected for the current 15-minute interval. The statistics are collected from the Far End Block Error (FEBE) code within the third 22 byte of the line overhead in Broadband ISDN (BISDN) applications. The definitions are the same as described for the near-end information.

The object identifiers are listed in Table 2-11.

Table 2-11 SONET/SDH Far End Line Current Table Object Identifiers

Name	Object Identifier
sonetFarEndLineCurrentTable	::= { sonetFarEndLine 1 }
sonetFarEndLineCurrentEntry	::= { sonetFarEndLineCurrentTable 1 }
sonetFarEndLineCurrentESs	::= { sonetFarEndLineCurrentEntry 1 }
sonetFarEndLineCurrentSESs	::= { sonetFarEndLineCurrentEntry 2 }
sonetFarEndLineCurrentCVs	::= { sonetFarEndLineCurrentEntry 3 }
sonetFarEndLineCurrentUASs	::= { sonetFarEndLineCurrentEntry 4 }

The MIB objects are listed in Table 2-12.

Table 2-12 SONET/SDH Far End Line Current Table MIB Objects

Name	Syntax	Description	Default Value
sonetFarEndLineCurrent Table	`SEQUENCE OF SonetFarEndLineCurrent Entry`	Describes the SONET/SDH far end line current table. Max-Access: not-accessible Status: current	none
sonetFarEndLineCurrent Entry	`SonetFarEndLineCurrent Entry`	Provides an entry in the SONET/SDH far end line current table. Max-Access: not-accessible Status: current The index is `ifIndex`.	none
sonetFarEndLineCurrent ESs	`PerfCurrentCount`	Specifies the counter associated with the number of Far End Errored Seconds encountered by a SONET/SDH interface in the current 15-minute interval. Max-Access: read-only Status: current	none
sonetFarEndLineCurrent SESs	`PerfCurrentCount`	Specifies the counter associated with the number of Far End Severely Errored Seconds encountered by a SONET/SDH medium, section, and line interface in the current 15-minute interval. Max-Access: read-only Status: current	none
sonetFarEndLineCurrent CVs	`PerfCurrentCount`	Specifies the counter associated with the number of Far End Coding Violations reported through the FEBE count, which is encountered by a SONET/SDH medium, section, and line interface in the current 15-minute interval. Max-Access: read-only Status: current	none
sonetFarEndLineCurrent UASs	`PerfCurrentCount`	Specifies the counter associated with the number of Far End Unavailable Seconds encountered by a SONET/SDH medium, section, and line interface in the current 15-minute interval. Max-Access: read-only Status: current	none

SONET/SDH Path Current Table

The SONET/SDH path current table contains various statistics that are collected for the current 15-minute interval.

The object identifiers are listed in Table 2-13.

Table 2-13 SONET/SDH Path Current Table Object Identifiers

Name	Object Identifier
sonetPathCurrentTable	::= { sonetPath 1 }
sonetPathCurrentEntry	::= { sonetPathCurrentTable 1 }
sonetPathCurrentWidth	::= { sonetPathCurrentEntry 1 }
sonetPathCurrentStatus	::= { sonetPathCurrentEntry 2 }
sonetPathCurrentESs	::= { sonetPathCurrentEntry 3 }
sonetPathCurrentSESs	::= { sonetPathCurrentEntry 4 }
sonetPathCurrentCVs	::= { sonetPathCurrentEntry 5 }
sonetPathCurrentUASs	::= { sonetPathCurrentEntry 6 }

The MIB objects are listed in Table 2-14.

Table 2-14 SONET/SDH Path Current Table MIB Objects

Name	Syntax	Description	Default Value
sonetPathCurrentTable	`SEQUENCE OF SonetPathCurrentEntry`	Describes the SONET/SDH path current table. Max-Access: not-accessible Status: current	none
sonetPathCurrentEntry	`SonetPathCurrentEntry`	Provides an entry in the SONET/SDH path current table. Max-Access: not-accessible Status: current The index is `ifIndex`.	none
sonetPathCurrentWidth	`INTEGER { sts1(1), sts3cSTM1(2), sts12cSTM4(3), sts24c(4), sts48cSTM16(5) }`	Indicates the type of the SONET/SDH path. For SONET, the assigned types are the `STS-Nc SPEs`. For example, *N* = `1`, `3`, `12`, `24`, and `48`. `STS-1` is equal to 51.84 Mbps. For SDH, the assigned types are the `STM-Nc VCs`. For example, *N* = `1`, `4`, and `16`. Max-Access: read-write Status: current	none

Table 2-14 SONET/SDH Path Current Table MIB Objects (continued)

<table>
<tr><th>Name</th><th>Syntax</th><th>Description</th><th>Default Value</th></tr>
<tr><td>sonetPathCurrentStatus</td><td>Integer32 (1..62)</td><td>Indicates the status of the interface. The sonetPathCurrentStatus is a bitmap represented as a sum; therefore, it can represent multiple defects simultaneously.
If no other flag is set, the sonetPathNoDefect is set.
Max-Access: read-only
Status: current
The following are the bit positions:
<table>
<tr><th>Bit Position</th><th>Variable</th></tr>
<tr><td>1</td><td>sonetPathNoDefect</td></tr>
<tr><td>2</td><td>sonetPathSTSLOP</td></tr>
<tr><td>4</td><td>sonetPathSTSAIS</td></tr>
<tr><td>8</td><td>sonetPathSTSRDI</td></tr>
<tr><td>16</td><td>sonetPathUnequipped</td></tr>
<tr><td>32</td><td>sonetPathSignalLabelMismatch</td></tr>
</table></td><td>none</td></tr>
<tr><td>sonetPathCurrentESs</td><td>PerfCurrentCount</td><td>Specifies the counter associated with the number of Errored Seconds encountered by a SONET/SDH path in the current 15-minute interval.
Max-Access: read-only
Status: current</td><td>none</td></tr>
<tr><td>sonetPathCurrentSESs</td><td>PerfCurrentCount</td><td>Specifies the counter associated with the number of severely Errored Seconds encountered by a SONET/SDH path in the current 15-minute interval.
Max-Access: read-only
Status: current</td><td>none</td></tr>
<tr><td>sonetPathCurrentCVs</td><td>PerfCurrentCount</td><td>Specifies the counter associated with the number of coding violations encountered by a SONET/SDH path in the current 15-minute interval.
Max-Access: read-only
Status: current</td><td>none</td></tr>
<tr><td>sonetPathCurrentUASs</td><td>PerfCurrentCount</td><td>Specifies the counter associated with the number of unavailable seconds encountered by a path in the current 15-minute interval.
Max-Access: read-only
Status: current</td><td>none</td></tr>
</table>

SONET/SDH Path Interval Table

The SONET/SDH path interval table contains various statistics that are collected by each system for the previous 24 hours. The past 24 hours is broken into 96 completed 15-minute intervals. Each system is required to store at least four completed 15 minute interval. The default value is 32 intervals.

The object identifiers are listed in Table 2-15.

Table 2-15 SONET/SDH Path Interval Table Object Identifiers

Name	Object Identifier
sonetPathIntervalTable	::= { sonetPath 2 }
sonetPathIntervalEntry	::= { sonetPathIntervalTable 1 }
sonetPathIntervalNumber	::= { sonetPathIntervalEntry 1 }
sonetPathIntervalESs	::= { sonetPathIntervalEntry 2 }
sonetPathIntervalSESs	::= { sonetPathIntervalEntry 3 }
sonetPathIntervalCVs	::= { sonetPathIntervalEntry 4 }
sonetPathIntervalUASs	:= { sonetPathIntervalEntry 5 }
sonetPathIntervalValidData	::= { sonetPathIntervalEntry 6 }

The MIB objects are listed in Table 2-16.

Table 2-16 SONET/SDH Path Interval Table MIB Objects

Name	Syntax	Description	Default Value
sonetPathIntervalTable	`SEQUENCE OF SonetPathIntervalEntry`	Describes the SONET/SDH path interval table. Max-Access: not-accessible Status: current	none
sonetPathIntervalEntry	`SonetPathIntervalEntry`	Provides an entry in the SONET/SDH path interval table. Max-Access: not-accessible Status: current The indexes include: • `ifIndex` • `sonetPathIntervalNumber`	none
sonetPathIntervalNumber	`Integer32 (1..96)`	Determines a number from 1 to 96 that identifies the interval for which the set of statistics is available. The interval identified by one is the most recently completed 15-minute interval, and the interval identified by `N` is the interval immediately preceding the one identified by `N-1`. Max-Access: not-accessible Status: current	none

Table 2-16 SONET/SDH Path Interval Table MIB Objects (continued)

Name	Syntax	Description	Default Value
sonetPathIntervalESs	PerfIntervalCount	Specifies the counter associated with the number of Errored Seconds encountered by a SONET/SDH path in a particular 15-minute interval in the past 24 hours. Max-Access: read-only Status: current	none
sonetPathIntervalSESs	PerfIntervalCount	Specifies the counter associated with the number of severely Errored Seconds encountered by a SONET/SDH path in a particular 15-minute interval in the past 24 hours. Max-Access: read-only Status: current	none
sonetPathIntervalCVs	PerfIntervalCount	Specifies the counter associated with the number of coding violations encountered by a SONET/SDH path in a particular 15-minute interval in the past 24 hours. Max-Access: read-only Status: current	none
sonetPathIntervalUASs	PerfIntervalCount	Specifies the counter associated with the number of unavailable seconds encountered by a SONET/SDH path in a particular 15-minute interval in the past 24 hours. Max-Access: read-only Status: current	none
sonetPathIntervalValid Data	TruthValue	Indicates if the data for this interval is valid. Max-Access: read-only Status: current	none

SONET/SDH Far End Path Current Table

The SONET/SDH far end path current table contains various statistics that are collected for the current 15-minute interval. The statistics are collected from the FEBE code within the G1 byte of the Path Overhead. The definitions are the same as described for the near-end information.

The object identifiers are listed in Table 2-17.

Table 2-17 SONET/SDH Far End Path Current Table Object Identifiers

Name	Object Identifier
sonetFarEndPathCurrentTable	::= { sonetFarEndPath 1 }
sonetFarEndPathCurrentEntry	::= { sonetFarEndPathCurrentTable 1 }
sonetFarEndPathCurrentESs	::= { sonetFarEndPathCurrentEntry 1 }
sonetFarEndPathCurrentSESs	::= { sonetFarEndPathCurrentEntry 2 }

Table 2-17 SONET/SDH Far End Path Current Table Object Identifiers (continued)

Name	Object Identifier
sonetFarEndPathCurrentCVs	::= { sonetFarEndPathCurrentEntry 3 }
sonetFarEndPathCurrentUASs	::= { sonetFarEndPathCurrentEntry 4 }

The MIB objects are listed in Table 2-18.

Table 2-18 SONET/SDH Far End Path Current Table MIB Objects

Name	Syntax	Description	Default Value
sonetFarEndPathCurrent Table	`SEQUENCE OF SonetFarEndPathCurrent Entry`	Describes the SONET/SDH far end path current table. Max-Access: not-accessible Status: current	none
sonetFarEndPathCurrent Entry	`SonetFarEndPathCurrent Entry`	Provides an entry in the SONET/SDH far end path current table. Max-Access: not-accessible Status: current The index is `ifIndex`.	none
sonetFarEndPathCurrent ESs	`PerfCurrentCount`	Specifies the counter associated with the number of Far End Errored Seconds encountered by a SONET/SDH interface in the current 15-minute interval. Max-Access: read-only Status: current	none
sonetFarEndPathCurrent SESs	`PerfCurrentCount`	Specifies the counter associated with the number of Far End Severely Errored Seconds encountered by a SONET/SDH path interface in the current 15-minute interval. Max-Access: read-only Status: current	none
sonetFarEndPathCurrent CVs	`PerfCurrentCount`	Specifies the counter associated with the number of Far End Coding Violations reported through the far end block error count. This object is encountered by a SONET/SDH path interface in the current 15-minute interval. Max-Access: read-only Status: current	none
sonetFarEndPathCurrent UASs	`PerfCurrentCount`	Specifies the counter associated with the number of Far End unavailable seconds encountered by a SONET/SDH path interface in the current 15-minute interval. Max-Access: read-only Status: current	none

SONET/SDH Far End Path Interval Table

The SONET/SDH Far End Path Interval Table contains various statistics collected by each system for the previous 24 hours. The past 24 hours is broken into 96 completed 15-minute intervals. Each system is required to store at least four completed 15-minute interval. The default value is 32 intervals.

The object identifiers are listed in Table 2-19.

Table 2-19 SONET/SDH Far End Path Interval Table Object Identifiers

Name	Object Identifier
sonetFarEndPathIntervalTable	::= { sonetFarEndPath 2 }
sonetFarEndPathIntervalEntry	::= { sonetFarEndPathIntervalTable 1 }
sonetFarEndPathIntervalNumber	::= { sonetFarEndPathIntervalEntry 1 }
sonetFarEndPathIntervalESs	::= { sonetFarEndPathIntervalEntry 2 }
sonetFarEndPathIntervalSESs	::= { sonetFarEndPathIntervalEntry 3 }
sonetFarEndPathIntervalCVs	::= { sonetFarEndPathIntervalEntry 4 }
sonetFarEndPathIntervalUASs	::= { sonetFarEndPathIntervalEntry 5 }
sonetFarEndPathIntervalValidData	::= { sonetFarEndPathIntervalEntry 6 }

The MIB objects are listed in Table 2-20.

Table 2-20 SONET/SDH Far End Path Interval Table MIB Objects

Name	Syntax	Description	Default Value
sonetFarEndPathInterval Table	`SEQUENCE OF SonetFarEndPathInterval Entry`	Describes the SONET/SDH far end path interval table. Max-Access: not-accessible Status: current	none
sonetFarEndPathInterval Entry	`SonetFarEndPathInterval Entry`	Provides an entry in the SONET/SDH far end path interval table. Max-Access: not-accessible Status: current The indexes include: • `ifIndex` • `sonetFarEndPathIntervalNumber`	none
sonetFarEndPathInterval Number	`Integer32 (1..96)`	Specifies a number from 1 to 96 that identifies the interval for which the set of statistics is available. The interval identified by one is the most recently completed 15-minute interval, and the interval identified by *N* is the interval immediately preceding the one identified by `N-1`. Max-Access: not-accessible Status: current	none

Table 2-20 SONET/SDH Far End Path Interval Table MIB Objects (continued)

Name	Syntax	Description	Default Value
sonetFarEndPathIntervalESs	PerfIntervalCount	Specifies the counter associated with the number of Far End Errored Seconds encountered by a SONET/SDH path interface in a particular 15-minute interval in the past 24 hours. Max-Access: read-only Status: current	none
sonetFarEndPathIntervalSESs	PerfIntervalCount	Specifies the counter associated with the number of Far End Severely Errored Seconds encountered by a SONET/SDH path interface in a particular 15-minute interval in the past 24 hours. Max-Access: read-only Status: current	none
sonetFarEndPathIntervalCVs	PerfIntervalCount	Specifies the counter associated with the number of Far End coding violations. This object is reported through the far end block error count, which is encountered by a SONET/SDH Path interface in a particular 15-minute interval in the past 24 hours. Max-Access: read-only Status: current	none
sonetFarEndPathIntervalUASs	PerfIntervalCount	Specifies the counter associated with the number of Far End unavailable seconds encountered by a SONET/SDH path interface in a particular 15-minute interval in the past 24 hours. Max-Access: read-only Status: current	none
sonetFarEndPathIntervalValidData	TruthValue	Indicates if the data for this interval is valid. Max-Access: read-only Status: current	none

SONET Conformance and Compliance Information

The object identifiers are listed in Table 2-21 for conformance.

Table 2-21 SONET Conformance Groups

Name	Object Identifier
sonetConformance	::= { sonetMIB 4 }
sonetGroups	::= { sonetConformance 1 }
sonetCompliances	::= { sonetConformance 2 }

SONET Compliance Statements

One object identifier is listed in Table 2-22 for compliance.

Table 2-22 SONET Compliance Object Identifier

Name	Object Identifier
sonetFarEndVTStuff2	::= { sonetCompliances 2 }

The objects used for compliance are listed in Table 2-23.

Table 2-23 SONET Compliance Objects

Name	Mandatory Groups	Description	Default Value
sonetCompliance2	`sonetMediumStuff2, sonetSectionStuff2`	Describes the compliance statement for the SONET/SDH interfaces. Status: current	none
sonetMediumType	none	Specifies that write access is not required. Min-Access: read-only	none
sonetMediumLineCoding	none	Specifies that write access is not required. Min-Access: read-only	none
sonetMediumLineType	none	Specifies that write access is not required. Min-Access: read-only	none
sonetMediumCircuitIdentifier	none	Specifies that write access is not required. Min-Access: read-only	none
sonetMediumLoopbackConfig	none	Specifies that write access is not required. Min-Access: read-only	none
sonetSESthresholdSet	none	Specifies that write access is not required. One enumerated value is supported. Min-Access: read-only	none
sonetLineStuff2	none	Specifies the implementation of this group is mandatory for all SONET/SDH systems that terminate SONET/SDH lines, paths, or virtual tributaries.	none
sonetFarEndLineStuff2	none	Specifies the implementation of this group is optional for all SONET/SDH systems that terminate SONET/SDH lines, paths, or virtual tributaries. It provides information for the Far End Block Error (FEBE) at the SONET/SDH line layer.	none
sonetPathStuff2	none	Specifies the implementation of this group is mandatory for all SONET/SDH systems that terminate SONET/SDH lines, paths, or virtual tributaries.	none
sonetPathCurrentWidth	none	Specifies that write access is not required. Min-Access: read-only	none

Table 2-23 SONET Compliance Objects (continued)

Name	Mandatory Groups	Description	Default Value
sonetFarEndPathStuff2	none	Specifies the implementation of this group is optional for all SONET/SDH systems that terminate SONET/SDH lines, paths, virtual tributaries, or process far end information.	none
sonetVTStuff2	none	Specifies the implementation of this group is mandatory for all SONET/SDH systems that terminate SONET/SDH virtual tributaries.	none
sonetVTCurrentWidth	none	Specifies that write access is not required. Min-Access: read-only	none
sonetFarEndVTStuff2	none	Specifies the implementation of this group is optional for all SONET/SDH systems that terminate SONET/SDH floating virtual tributaries or process far end information.	none

SONET Units of Conformance

The object identifiers for SONET are listed in Table 2-24.

Table 2-24 SONET Units of Conformance Object Identifiers

Name	Object Identifier
sonetMediumStuff2	::= { sonetGroups 9 }
sonetSectionStuff2	::= { sonetGroups 10 }
sonetLineStuff2	::= { sonetGroups 11 }
sonetPathStuff2	::= { sonetGroups 12 }
sonetVTStuff2	::= { sonetGroups 13 }
sonetFarEndLineStuff2	::= { sonetGroups 14 }
sonetFarEndPathStuff2	::= { sonetGroups 15 }
sonetFarEndVTStuff2	::= { sonetGroups 16 }

The objects are listed in Table 2-25 for units of conformance.

Table 2-25 SONET Units of Conformance Objects

Name	Objects	Description	Default Value
sonetMediumStuff2	sonetMediumType, sonetMediumTimeElapsed, sonetMediumValidIntervals, sonetMediumLineCoding, sonetMediumLineType, sonetMediumCircuitIdentifier, sonetMediumInvalidIntervals, sonetMediumLoopbackConfig, sonetSESthresholdSet	Provides configuration information applicable to all SONET/SDH interfaces for a collection of objects. Status: current	none
sonetSectionStuff2	sonetSectionCurrentStatus, sonetSectionCurrentESs, sonetSectionCurrentSESs, sonetSectionCurrentSEFSs, sonetSectionCurrentCVs, sonetSectionIntervalESs, sonetSectionIntervalSESs, sonetSectionIntervalSEFSs, sonetSectionIntervalCVs, sonetSectionIntervalValidData	Provides information specific to SONET/SDH section interfaces for a collection of objects. Status: current	none
sonetLineStuff2	sonetLineCurrentStatus, sonetLineCurrentESs, sonetLineCurrentSESs, sonetLineCurrentCVs, sonetLineCurrentUASs, sonetLineIntervalESs, sonetLineIntervalSESs, sonetLineIntervalCVs, sonetLineIntervalUASs, sonetLineIntervalValidData	Provides information specific to SONET/SDH line interfaces for a collection of objects. Status: current	none
sonetPathStuff2	sonetPathCurrentWidth, sonetPathCurrentStatus, sonetPathCurrentESs, sonetPathCurrentSESs, sonetPathCurrentCVs, sonetPathCurrentUASs, sonetPathIntervalESs, sonetPathIntervalSESs, sonetPathIntervalCVs, sonetPathIntervalUASs, sonetPathIntervalValidData	Provides information specific to SONET/SDH path interfaces for a collection of objects. Status: current	none
sonetVTStuff2	sonetVTCurrentWidth, sonetVTCurrentStatus, sonetVTCurrentESs, sonetVTCurrentSESs, sonetVTCurrentCVs, sonetVTCurrentUASs, sonetVTIntervalESs, sonetVTIntervalSESs, sonetVTIntervalCVs, sonetVTIntervalUASs, sonetVTIntervalValidData	Provides information specific to SONET/SDH Virtual Tributary (VT) interfaces for a collection of objects. Status: current	none

Table 2-25 SONET Units of Conformance Objects (continued)

Name	Objects	Description	Default Value
sonetFarEndLineStuff2	sonetFarEndLineCurrentESs, sonetFarEndLineCurrentSESs, sonetFarEndLineCurrentCVs, sonetFarEndLineCurrentUASs, sonetFarEndLineIntervalESs, sonetFarEndLineIntervalSESs, sonetFarEndLineIntervalCVs, sonetFarEndLineIntervalUASs, sonetFarEndLineIntervalValidData	Provides information specific to SONET/SDH line interfaces, and maintains Line Far End information for a collection of objects. Status: current	none
sonetFarEndPathStuff2	sonetFarEndPathCurrentESs, sonetFarEndPathCurrentSESs, sonetFarEndPathCurrentCVs, sonetFarEndPathCurrentUASs, sonetFarEndPathIntervalESs, sonetFarEndPathIntervalSESs, sonetFarEndPathIntervalCVs, sonetFarEndPathIntervalUASs, sonetFarEndPathIntervalValidData	Provides information specific to SONET/SDH path interfaces, and maintains Path Far End information for a collection of objects. Status: current	none
sonetFarEndVTStuff2	sonetFarEndVTCurrentESs, sonetFarEndVTCurrentSESs, sonetFarEndVTCurrentCVs, sonetFarEndVTCurrentUASs, sonetFarEndVTIntervalESs, sonetFarEndVTIntervalSESs, sonetFarEndVTIntervalCVs, sonetFarEndVTIntervalUASs, sonetFarEndVTIntervalValidData	Provides information specific to SONET/SDH VT interfaces, and maintains VT Far End information for a collection of objects. Status: current	none

DS3/E3 MIB Objects

This MIB module is extracted from RFC 2496, which describes DS3 and E3 interface objects. DS3/E3 MIB objects reside in the DS3-MIB.my file.

Note The DS3 MIB is supported by AXSM.

The DS/E3 MIB objects include:

- DS3/E3 Near End Group
- DS3 Far End Group
- ds3 Traps
- DS/E3 Conformance Information

DS3/E3 Near End Group

The DS3/E3 near end group consists of four tables:

- DS3/E3 Configuration Table
- DS3/E3 Current Table
- DS3/E3 Interval Table
- DS3/E3 Total

DS3/E3 Configuration Table

The object identifier for each MIB object is listed in Table 2-26.

Table 2-26 DS3/E3 Configuration Table Object Identifiers

Name	Object Identifier
dsx3ConfigTable	::= {ds3 5}
dsx3ConfigEntry	::= {dsx3ConfigTable 1}
dsx3LineIndex	::= {dsx3ConfigEntry 1}
dsx3IfIndex	::= {dsx3ConfigEntry 2}
dsx3TimeElapsed	::= {dsx3ConfigEntry 3}
dsx3ValidIntervals	::= {dsx3ConfigEntry 4}
dsx3LineType	::= {dsx3ConfigEntry 5}
dsx3LineCoding	::= {dsx3ConfigEntry 6}
dsx3SendCode	::= {dsx3ConfigEntry 7}
dsx3CircuitIdentifier	::= {dsx3ConfigEntry 8}
dsx3LoopbackConfig	::= {dsx3ConfigEntry 9}
dsx3LineStatus	::= {dsx3ConfigEntry 10}
dsx3TransmitClockSource	::= {dsx3ConfigEntry 11}

Table 2-26 DS3/E3 Configuration Table Object Identifiers (continued)

Name	Object Identifier
dsx3InvalidIntervals	::= {dsx3ConfigEntry 12}
dsx3LineLength	::= {dsx3ConfigEntry 13}
dsx3LineStatusLastChange	::= {dsx3ConfigEntry 14}
dsx3LineStatusChangeTrapEnable	::= {dsx3ConfigEntry 15}
dsx3LoopbackStatus	::= {dsx3ConfigEntry 16}
dsx3Channelization	::= {dsx3ConfigEntry 17}
dsx3Ds1ForRemoteLoop	::= {dsx3ConfigEntry 18}

The MIB objects are listed in Table 2-27.

Table 2-27 DS3/E3 Configuration Table MIB Objects

Name	Syntax	Description	Default Value
dsx3ConfigTable	`Sequence of Dsx3ConfigEntry`	Describes the DS3/E3 configuration table. Max-Access: not-accessible Status: current	none
dsx3ConfigEntry	`dsx3ConfigEntry`	Provides an entry in the DS3/E3 configuration table. Max-Access: not-accessible Status: current The index is `dsx3LineIndex`.	none
dsx3LineIndex	`InterfaceIndex`	Equals `ifIndex`. `ds3xIfIndex` describes its previous usage. By making the object equal to `ifIndex`, it allows proper use of `ifStackTable`. Previously, this object was the identifier of a DS3/E3 Interface on a managed device. If there is an ifEntry that is directly associated with this and only this DS3/E3 interface, it can have the same value as `ifIndex`. Otherwise, number the dsx3LineIndices with a unique identifier following the rules of choosing a number that is greater than `ifNumber` and numbering the inside interfaces (for example, equipment side) with even numbers, and outside interfaces (for example, network side) with odd numbers. Max-Access: read-only Status: current	none
dsx3IfIndex	`InterfaceIndex`	Specifies the value for this object is equal to the value of `ifIndex` from the Interfaces table of MIB II (RFC 1213). Max-Access: read-only Status: deprecated	none

Table 2-27 DS3/E3 Configuration Table MIB Objects (continued)

<table>
<tr><th>Name</th><th>Syntax</th><th>Description</th><th>Default Value</th></tr>
<tr><td>dsx3TimeElapsed</td><td>INTEGER (0..899)</td><td>Determines the number of seconds that have elapsed since the beginning of the near end current error-measurement period. If, for some reason, such as an adjustment in the system time-of-day clock, the current interval exceeds the maximum value, the agent will return the maximum value.
Max-Access: read-only
Status: current</td><td>none</td></tr>
<tr><td>dsx3ValidIntervals</td><td>INTEGER (0..96)</td><td>Determines the number of previous near end intervals for which data was collected. The value is 96 unless the interface was brought online within the last 24 hours, in which case the value will be the number of complete 15-minute near end intervals since the interface has been online. In the case where the agent is a proxy, some intervals may not be available. In this case, this interval is the maximum interval number for which data is available.
Max-Access: read-only
Status: current</td><td>none</td></tr>
<tr><td>dsx3LineType</td><td>INTEGER {dsx3other(1), dsx3M23(2), dsx3SYNTRAN(3), dsx3CbitParity(4), dsx3ClearChannel(5), e3other(6), e3Framed(7), e3Plcp(8)}</td><td>Indicates the variety of DS3 C-bit or E3 application implementing this interface. The type of interface affects the interpretation of usage and error statistics. The rate of DS3 is 44.736 Mbps and E3 is 34.368 Mbps. The dsx3ClearChannel value means that the C-bits are not used except for sending/receiving Alarm Indication Signal (AIS).
Max-Access: read-write
Status: current
The specifications for each line type include:
<table>
<tr><th>dsx3 Line Type</th><th>Specification</th></tr>
<tr><td>dsx3M23</td><td>ANSI T1.107-1988 [9]</td></tr>
<tr><td>dsx3SYNTRAN</td><td>ANSI T1.107-1988 [9]</td></tr>
<tr><td>dsx3CbitParity</td><td>ANSI T1.107a-1990 [9a]</td></tr>
<tr><td>dsx3Clear Channel</td><td>ANSI T1.102-1987 [8]</td></tr>
<tr><td>e3Framed</td><td>CCITT G.751 [12]</td></tr>
<tr><td>e3Plcp</td><td>ETSI T/NA(91)18 [13]</td></tr>
</table></td><td>none</td></tr>
<tr><td>dsx3LineCoding</td><td>INTEGER {dsx3Other(1), dsx3B3ZS(2), e3HDB3(3)}</td><td>Describes the variety of Zero Code Suppression used on this interface, which in turn affects a number of its characteristics. dsx3B3ZS and e3HDB3 refer to the specified patterns of normal bits and bipolar violations, which are used to replace sequences of 0 bits of a specified length.
Max-Access: read-write
Status: current</td><td>none</td></tr>
</table>

Table 2-27 DS3/E3 Configuration Table MIB Objects (continued)

<table>
<tr><th>Name</th><th>Syntax</th><th>Description</th><th>Default Value</th></tr>
<tr><td>dsx3SendCode</td><td>INTEGER {dsx3SendNoCode(1), dsx3SendLineCode(2), dsx3SendPayloadCode(3), dsx3SendResetCode(4), dsx3SendDS1Loop Code(5), dsx3SendTestPattern(6)}</td><td>Indicates the type of code the device is sending across the DS3/E3 interface.
Note These are optional for E3 interfaces.
Setting this variable causes the interface to begin sending the code requested.
Max-Access: read-write
Status: current</td><td>none</td></tr>
<tr><td>dsx3Circuit Identifier</td><td>DisplayString (SIZE (0..255))</td><td>Contains the transmission vendor circuit identifier, which facilitates troubleshooting.
Max-Access: read-write
Status: current</td><td>none</td></tr>
<tr><td>dsx3Loopback Config</td><td>INTEGER {dsx3NoLoop(1), dsx3PayloadLoop(2), dsx3LineLoop(3), dsx3OtherLoop(4), dsx3InwardLoop(5), dsx3DualLoop(6)}</td><td>Represents the desired loopback configuration of the DS3/E3 interface.
Max-Access: read-write
Status: current
The definitions of each value include:
<table>
<tr><th>dsx3 Loopback Configuration</th><th>Definition</th></tr>
<tr><td>dsx3NoLoop</td><td>Not in the loopback state. A device that is not capable of performing a loopback on the interface shall always return back the value.</td></tr>
<tr><td>dsx3Payload Loop</td><td>The received signal at this interface is looped through the device. Typically the received signal is looped back for retransmission after it has passed through the device framing function.</td></tr>
<tr><td>dsx3LineLoop</td><td>The received signal at this interface does not go through the device (minimum penetration) but is looped back out.</td></tr>
<tr><td>dsx3OtherLoop</td><td>Loopbacks that are not defined here.</td></tr>
<tr><td>dsx3InwardLoop</td><td>The sent signal at this interface is looped back through the device.</td></tr>
<tr><td>dsx3DualLoop</td><td>Both dsx1LineLoop and dsx1InwardLoop are active simultaneously.</td></tr>
</table></td><td>none</td></tr>
</table>

Table 2-27 DS3/E3 Configuration Table MIB Objects (continued)

<table>
<tr><th>Name</th><th>Syntax</th><th>Description</th><th>Default Value</th></tr>
<tr><td>dsx3LineStatus</td><td>INTEGER (1..4095)</td><td>Indicates the line status of the interface. It contains loopback state information and failure state information. The dsx3LineStatus is a bitmap represented as a sum, therefore, it can represent multiple failures and a loopback (see the dsx3LoopbackConfig object for the type of loopback) simultaneously. The dsx3NoAlarm must be set if no other flag is set.

If the dsx3loopbackState bit is set, the loopback in effect can be determined from the dsx3loopbackConfig object.

Max-Access: read-only

Status: current

The various bit positions include:
<table>
<tr><th>Bit Position</th><th>Line Status</th><th>Meaning</th></tr>
<tr><td>1</td><td>dsx3NoAlarm</td><td>No alarm present</td></tr>
<tr><td>2</td><td>dsx3RcvRAIFailure</td><td>Receiving Yellow/Remote Alarm Indication</td></tr>
<tr><td>4</td><td>dsx3XmitRAIAlarm</td><td>Transmitting Yellow/Remote Alarm Indication</td></tr>
<tr><td>8</td><td>dsx3RcvAIS</td><td>Receiving AIS failure state</td></tr>
<tr><td>16</td><td>dsx3XmitAIS</td><td>Transmitting AIS</td></tr>
<tr><td>32</td><td>dsx3LOF</td><td>Receiving LOF failure state</td></tr>
<tr><td>64</td><td>dsx3LOS</td><td>Receiving LOS failure state</td></tr>
<tr><td>128</td><td>dsx3LoopbackState</td><td>Looping the received signal</td></tr>
<tr><td>256</td><td>dsx3RcvTestCode</td><td>Receiving a Test Pattern</td></tr>
<tr><td>512</td><td>dsx3OtherFailure</td><td>Any line status not defined here</td></tr>
<tr><td>1024</td><td>dsx3UnavailSig State</td><td>Near End in Unavailable Signal State</td></tr>
<tr><td>2048</td><td>dsx3NetEquipOOS</td><td>Carrier Equipment Out of Service</td></tr>
</table></td><td>none</td></tr>
</table>

Table 2-27 DS3/E3 Configuration Table MIB Objects (continued)

Name	Syntax	Description	Default Value
dsx3Transmit ClockSource	`INTEGER {loopTiming(1), localTiming(2), throughTiming(3)}`	Describes the source of Transmit Clock. The sources include: • `loopTiming`—The recovered receive clock is used as the transmit clock. • `localTiming`—A local clock source is used or that an external clock is attached to the box containing the interface. • `throughTiming`—Transmit clock is derived from the recovered receive clock of another DS3 interface. Max-Access: read-write Status: current	none
dsx3Invalid Intervals	`INTEGER (0..96)`	Determines the number of intervals in the range from 0 to `dsx3ValidIntervals` for which no data is available. This object is typically 0, except in cases where the data for some intervals are not available, for example, in proxy situations. Max-Access: read-only Status: current	none
dsx3LineLength	`INTEGER (0..64000) (in units of meters)`	Determines the length of the DS3 line in meters. This object provides information for line build out circuitry. Max-Access: read-write Status: current	none
dsx3LineStatusLast Change	`TimeStamp`	Determines the value of the MIB IF `sysUpTime` object at the time this DS3/E3 entered its current line status. If the current state was entered prior to the last reinitialization of the proxy agent, this object contains a 0 value. Max-Access: read-only Status: current	none
dsx3LineStatus ChangeTrapEnable	`INTEGER {enabled(1), disabled(2)}`	Indicates whether `dsx3LineStatusChange` traps are generated for this interface. Max-Access: read-write Status: current	`disabled`

Table 2-27 DS3/E3 Configuration Table MIB Objects (continued)

<table>
<tr><th>Name</th><th>Syntax</th><th>Description</th><th>Default Value</th></tr>
<tr><td>dsx3Loopback Status</td><td>INTEGER (1..127)</td><td>Represents the current state of the loopback on the DS3 interface. The variable contains information about loopbacks established by a manager remotely from the far end. The dsx3LoopbackStatus is a bitmap represented as a sum; therefore, it can represent multiple loopbacks simultaneously.
Max-Access: read-only
Status: current
The bit positions include:
<table>
<tr><th>Bit Position</th><th>Variable</th></tr>
<tr><td>1</td><td>dsx3NoLoopback</td></tr>
<tr><td>2</td><td>dsx3NearEndPayloadLoopback</td></tr>
<tr><td>4</td><td>dsx3NearEndLineLoopback</td></tr>
<tr><td>8</td><td>dsx3NearEndOtherLoopback</td></tr>
<tr><td>16</td><td>dsx3NearEndInwardLoopback</td></tr>
<tr><td>32</td><td>dsx3FarEndPayloadLoopback</td></tr>
<tr><td>64</td><td>dsx3FarEndLineLoopback</td></tr>
</table></td><td>none</td></tr>
<tr><td>dsx3Channelization</td><td>INTEGER {disabled(1), enabledDs1(2), enabledDs2(3)}</td><td>Indicates whether this DS3/E3 is channelized or unchannelized. The value of enabledDs1 indicates that this is a DS3 channelized into DS1s. The value of enabledDs3 indicates that this is a DS3 channelized into DS2s. Setting this object causes the creation or deletion of DS2 or DS1 entries in the ifTable.
Max-Access: read-write
Status: current</td><td>none</td></tr>
<tr><td>dsx3Ds1ForRemote Loop</td><td>INTEGER (0..29)</td><td>Indicates which DS1/E1 on this DS3/E3 is indicated in the remote DS1 loopback request. A value of 0 means no DS1 is looped. A value of 29 means all DS1/E1s are looped.
Max-Access: read-write
Status: current</td><td>none</td></tr>
</table>

DS3/E3 Current Table

The object identifier for each MIB object is listed in Table 2-28.

Table 2-28 DS3/E3 Current Table Object Identifiers

Name	Object Identifier
dsx3CurrentTable	::= {ds3 6}
dsx3CurrentEntry	::= {dsx3CurrentTable 1}
dsx3CurrentIndex	::= {dsx3CurrentEntry 1}
dsx3CurrentPESs	::= {dsx3CurrentEntry 2}
dsx3CurrentPSESs	::= {dsx3CurrentEntry 3}
dsx3CurrentSEFSs	::= {dsx3CurrentEntry 4}
dsx3CurrentUASs	::= {dsx3CurrentEntry 5}
dsx3CurrentLCVs	::= {dsx3CurrentEntry 6}
dsx3CurrentPCVs	::= {dsx3CurrentEntry 7}
dsx3CurrentLESs	::= {dsx3CurrentEntry 8}
dsx3CurrentCCVs	::= {dsx3CurrentEntry 9}
dsx3CurrentCESs	::= {dsx3CurrentEntry 10}
dsx3CurrentCSESs	::= {dsx3CurrentEntry 11}

The MIB objects are listed in Table 2-29.

Table 2-29 DS3/E3 Current Table MIB Objects

Name	Syntax	Description	Default Value
dsx3CurrentTable	`Sequence of dsx3CurrentEntry`	Contains various statistics being collected for the current 15-minute interval that are used for the DS/E3 current table. Max-Access: not-accessible Status: current	none
dsx3CurrentEntry	`dsx3CurrentEntry`	Provides the entry in the DS3/E3 Current table. Max-Access: not-accessible Status: current The index is `dsx3CurrentIndex`.	none
dsx3CurrentIndex	`InterfaceIndex`	Specifies the index value which uniquely identifies the DS3/E3 interface to which this entry is applicable. The interface is identified by a particular value of this index. It is the same interface that is identified by the same value for an `dsx3LineIndex` object instance. Max-Access: read-only Status: current	none

Table 2-29 DS3/E3 Current Table MIB Objects (continued)

Name	Syntax	Description	Default Value
dsx3CurrentPESs	PerfCurrentCount	Describes the counter associated with the number of P-bit Errored Seconds. Max-Access: read-only Status: current	none
dsx3CurrentPSESs	PerfCurrentCount	Describes the counter associated with the number of P-bit Severely Errored Seconds (PSES). Max-Access: read-only Status: current	none
dsx3CurrentSEFSs	PerfCurrentCount	Describes the counter associated with the number of Severely Errored Framing Seconds (SEFS). Max-Access: read-only Status: current	none
dsx3CurrentUASs	PerfCurrentCount	Describes the counter associated with the number of Unavailable Seconds (UAS). Max-Access: read-only Status: current	none
dsx3CurrentLCVs	PerfCurrentCount	Describes the counter associated with the number of Line Coding Violations (LCV). Max-Access: read-only Status: current	none
dsx3CurrentPCVs	PerfCurrentCount	Describes the counter associated with the number of P-bit Coding Violations (PCV). Max-Access: read-only Status: current	none
dsx3CurrentLESs	PerfCurrentCount	Describes the number of Line Errored Seconds (LES). Max-Access: read-only Status: current	none
dsx3CurrentCCVs	PerfCurrentCount	Describes the number of C-bit Coding Violations (CV). Max-Access: read-only Status: current	none
dsx3CurrentCESs	PerfCurrentCount	Describes the number of C-bit Errored Seconds (CES). Max-Access: read-only Status: current	none
dsx3CurrentCSESs	PerfCurrentCount	Describes the number of C-bit Severely Errored Seconds (CSES). Max-Access: read-only Status: current	none

DS3/E3 Interval Table

The object identifiers are listed in Table 2-30.

Table 2-30 DS3/E3 Interval Table Object Identifiers

Name	Object Identifier
dsx3IntervalTable	::= { ds3 7 }
dsx3IntervalEntry	::= { dsx3IntervalTable 1 }
dsx3IntervalIndex	::= {dsx3IntervalEntry 1}
dsx3IntervalNumber	::= {dsx3IntervalEntry 2}
dsx3IntervalPESs	::= {dsx3IntervalEntry 3}
dsx3IntervalPSESs	::= {dsx3IntervalEntry 4}
dsx3IntervalSEFSs	::= {dsx3IntervalEntry 5}
dsx3IntervalUASs	::= {dsx3IntervalEntry 6}
dsx3IntervalLCVs	::= {dsx3IntervalEntry 7}
dsx3IntervalPCVs	::= {dsx3IntervalEntry 8}
dsx3IntervalLESs	::= {dsx3IntervalEntry 9}
dsx3IntervalCCVs	::= {dsx3IntervalEntry 10}
dsx3IntervalCESs	::= {dsx3IntervalEntry 11}
dsx3IntervalCSESs	::= {dsx3IntervalEntry 12}
dsx3IntervalValidData	::= {dsx3IntervalEntry 13}

The MIB objects are listed in Table 2-31.

Table 2-31 DS3/E3 Interval Table MIB Objects

Name	Syntax	Description	Default Value
dsx3IntervalTable	`Sequence of dsx3IntervalEntry`	Contains various statistics collected by each DS3/E3 interface over the previous 24 hours of operation for the DS3/E3 interval table. 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`). Max-Access: not-accessible Status: current	none
dsx3IntervalEntry	`dsx3IntervalEntry`	Provides an entry in the DS3/E3 interval table. Max-Access: not-accessible Status: current The indexes include: • `dsx3IntervalIndex` • `dsx3IntervalNumber`	none
dsx3IntervalIndex	`InterfaceIndex`	Specifies the index value which uniquely identifies the DS3/E3 interface to which this entry applies. The interface identified by a particular value of this index is the same interface as identified by the same value an `dsx3LineIndex` object instance. Max-Access: read-only Status: current	none
dsx3IntervalNumber	`INTEGER (1..96)`	Determines a number from 1 to 96. For example, 1 is the most recently completed 15-minute interval. 96 is the 15-minute interval completed for 23 hours and 45 minutes prior to interval 1. Max-Access: read-only Status: current	none
dsx3IntervalPESs	`PerfIntervalCount`	Describes the counter associated with the number of P-bit Errored Seconds (PES). Max-Access: read-only Status: current	none
dsx3IntervalPSESs	`PerfIntervalCount`	Describes the counter associated with the number of P-bit Severely Errored Seconds (PSES). Max-Access: read-only Status: current	none

Table 2-31 DS3/E3 Interval Table MIB Objects (continued)

Name	Syntax	Description	Default Value
dsx3IntervalSEFSs	PerfIntervalCount	Describes the counter associated with the number of Severely Errored Framing Seconds (SEFS). Max-Access: read-only Status: current	none
dsx3IntervalUASs	PerfIntervalCount	Describes the counter associated with the number of Unavailable Seconds (UA). This object can decrease if the unavailable seconds occur across an interval boundary. Max-Access: read-only Status: current	none
dsx3IntervalLCVs	PerfIntervalCount	Describes the counter associated with the number of Line Coding Violations (LCV). Max-Access: read-only Status: current	none
dsx3IntervalPCVs	PerfIntervalCount	Describes the counter associated with the number of P-bit Coding Violations (PCV). Max-Access: read-only Status: current	none
dsx3IntervalLESs	PerfIntervalCount	Determines the number of Line Errored Seconds (LES) (BPVs or illegal 0 sequences). Max-Access: read-only Status: current	none
dsx3IntervalCCVs	PerfIntervalCount	Determines the number of C-bit Coding Violations (CCV). Max-Access: read-only Status: current	none
dsx3IntervalCESs	PerfIntervalCount	Determines the number of C-bit Errored Seconds (CES). Max-Access: read-only Status: current	none
dsx3IntervalCSESs	PerfIntervalCount	Determines the number of C-bit Severely Errored Seconds (CSES). Max-Access: read-only Status: current	none
dsx3IntervalValidData	TruthValue	Indicates if the data for this interval is valid. Max-Access: read-only Status: current	none

DS3/E3 Total

The object identifiers are listed in Table 2-32.

Table 2-32 DS3/E3 Total Object Identifiers

Name	Object Identifier
dsx3TotalTable	::= {ds3 8}
dsx3TotalEntry	::= {dsx3TotalTable 1}
dsx3TotalIndex	::= {dsx3TotalEntry 1}
dsx3TotalPESs	::= {dsx3TotalEntry 2}
dsx3TotalPSESs	::= {dsx3TotalEntry 3}
dsx3TotalSEFSs	::= {dsx3TotalEntry 4}
dsx3TotalUASs	::= {dsx3TotalEntry 5}
dsx3TotalLCVs	::= {dsx3TotalEntry 6}
dsx3TotalPCVs	::= {dsx3TotalEntry 7}
dsx3TotalLESs	::= {dsx3TotalEntry 8}
dsx3TotalCCVs	::= {dsx3TotalEntry 9}
dsx3TotalCESs	::= {dsx3TotalEntry 10}
dsx3TotalCSESs	::= {dsx3TotalEntry 11}

The MIB objects are listed in Table 2-33.

Table 2-33 DS3/E3 Total MIB Objects

Name	Syntax	Description	Default Value
dsx3TotalTable	`Sequence of dsx3TotalEntry`	Contains the sum of the various statistics for the 24-hour period preceding the current interval. Max-Access: not-accessible Status: current	none
dsx3TotalEntry	`dsx3TotalEntry`	Provides an entry in the DS3/E3 Total table. Max-Access: not-accessible Status: current The index is `dsx3TotalIndex`.	none
dsx3TotalIndex	`InterfaceIndex`	Specifies the index value which uniquely identifies the DS3/E3 interface to which this entry applies. The interface identified by a particular value of this index is the same interface as identified by the same value as an `dsx3LineIndex` object instance. Max-Access: read-only Status: current	none

Table 2-33 DS3/E3 Total MIB Objects (continued)

Name	Syntax	Description	Default Value
dsx3TotalPESs	PerfTotalCount	Describes the counter associated with the number of P-bit Errored Seconds (PES) encountered by a DS3 interface in the previous 24-hour interval. Invalid 15-minute intervals count as 0. Max-Access: read-only Status: current	none
dsx3TotalPSESs	PerfTotalCount	Describes the counter associated with the number of P-bit Severely Errored Seconds (PSES) encountered by a DS3 interface in the previous 24-hour interval. Invalid 15-minute intervals count as 0. Max-Access: read-only Status: current	none
dsx3TotalSEFSs	PerfTotalCount	Describes the counter associated with the number of Severely Errored Framing Seconds (SEFS) encountered by a DS3/E3 interface in the previous 24-hour interval. Invalid 15-minute intervals count as 0. Max-Access: read-only Status: current	none
dsx3TotalUASs	PerfTotalCount	Describes the counter associated with the number of Unavailable Seconds (UA) encountered by a DS3 interface in the previous 24-hour interval. Invalid 15-minute intervals count as 0. Max-Access: read-only Status: current	none
dsx3TotalLCVs	PerfTotalCount	Describes the counter associated with the number of Line Coding Violations (LCV) encountered by a DS3/E3 interface in the previous 24-hour interval. Invalid 15-minute intervals count as 0. Max-Access: read-only Status: current	none
dsx3TotalPCVs	PerfTotalCount	Describes the counter associated with the number of P-bit Coding Violations (PCV) encountered by a DS3 interface in the previous 24-hour interval. Invalid 15-minute intervals count as 0. Max-Access: read-only Status: current	none

Table 2-33 DS3/E3 Total MIB Objects (continued)

Name	Syntax	Description	Default Value
dsx3TotalLESs	PerfTotalCount	Determines the number of Line Errored Seconds (LES) (BPVs or illegal 0 sequences) encountered by a DS3/E3 interface in the previous 24-hour interval. Invalid 15-minute intervals count as 0. Max-Access: read-only Status: current	none
dsx3TotalCCVs	PerfToCount	Determines the number of C-bit Coding Violations (CCV) encountered by a DS3 interface in the previous 24-hour interval. Invalid 15-minute intervals count as 0. Max-Access: read-only Status: current	none
dsx3TotalCESs	PerfTotalCount	Determines the number of C-bit Errored Seconds (CES) encountered by a DS3 interface in the previous 24-hour interval. Invalid 15-minute intervals count as 0. Max-Access: read-only Status: current	none
dsx3TotalCSESs	PerfTotalCount	Determines the number of C-bit Severely Errored Seconds (CSES) encountered by a DS3 interface in the previous 24-hour interval. Invalid 15-minute intervals count as 0. Max-Access: read-only Status: current	none

DS3 Far End Group

The DS3 Far End Group consists of four tables:

- DS3 Far End Configuration Table
- DS3 Far End Current Table
- DS3 Far End Interval Table
- DS3 Far End Total Table

DS3 Far End Configuration Table

The object identifiers are listed in Table 2-34.

Table 2-34 DS3/E3 Far End Configuration Table Object Identifiers

Name	Object Identifier
dsx3FarEndConfigTable	::= {ds3 9}
dsx3FarEndConfigEntry	::= {dsx3FarEndConfigTable 1}
dsx3FarEndLineIndex	:= {dsx3FarEndConfigEntry 1}
dsx3FarEndEquipCode	::= {dsx3FarEndConfigEntry 2}
dsx3FarEndLocationIDCode	::= {dsx3FarEndConfigEntry 3}
dsx3FarEndFrameIDCode	::= {dsx3FarEndConfigEntry 4}
dsx3FarEndUnitCode	::= {dsx3FarEndConfigEntry 5}
dsx3FarEndFacilityIDCode	::= {dsx3FarEndConfigEntry 6}

The MIB objects are listed in Table 2-35.

Table 2-35 DS3/E3 Far End Configuration Table MIB Objects

Name	Syntax	Description	Default Value
dsx3FarEndConfig Table	`Sequence of dsx3FarEndConfigEntry`	Contains the configuration information reported in the C-bits from the remote end. Max-Access: not-accessible Status: current	none
dsx3FarEndConfig Entry	`dsx3FarEndConfigEntry`	Provides an entry in the DS3 Far End configuration table. Max-Access: not-accessible Status: current The index is `dsx3FarEndLineIndex`.	none
dsx3FarEndLineIndex	`Interface Index`	Specifies the index value which uniquely identifies the DS3 interface to which this entry applies. The interface identified by a particular value of this index is the same interface as identified by the same value an `dsx3LineIndex` object instance. Max-Access: read-only Status: current	none
dsx3FarEndEquip Code	`DisplayString (SIZE (0..10))`	Describes the specific piece of equipment. The Far End Equipment Identification code is sent within the Path Identification Message. Max-Access: read-write Status: current	none

Table 2-35 DS3/E3 Far End Configuration Table MIB Objects (continued)

Name	Syntax	Description	Default Value
dsx3FarEndLocation IDCode	DisplayString (SIZE (0..11))	Describes the specific location of the equipment. The Far End Location Identification code is sent within the Path Identification Message. Max-Access: read-write Status: current	none
dsx3FarEndFrameID Code	DisplayString (SIZE (0..10))	Identifies where the equipment is located within a building. The Far End Frame Identification code is sent within the Path Identification Message. Max-Access: read-write Status: current	none
dsx3FarEndUnitCode	DisplayString (SIZE (0..6))	Identifies the equipment location within a bay. The Far End code is sent within the Path Identification Message. Max-Access: read-write Status: current	none
dsx3FarEndFacilityID Code	DisplayString (SIZE (0..38))	Identifies a specific Far End DS3 path. The Far End code is sent within the Path Identification Message. Max-Access: read-write Status: current	none

DS3 Far End Current Table

The object identifiers are listed in Table 2-36.

Table 2-36 DS3/E3 Far End Current Table Object Identifiers

Name	Object Identifier
dsx3FarEndCurrentTable	::= {ds3 10}
dsx3FarEndCurrentEntry	::= {dsx3FarEndCurrentTable 1}
dsx3FarEndCurrentIndex	::= {dsx3FarEndCurrentEntry 1}
dsx3FarEndTimeElapsed	::= {dsx3FarEndCurrentEntry 2}
dsx3FarEndValidIntervals	::= {dsx3FarEndCurrentEntry 3}
dsx3FarEndCurrentCESs	::= {dsx3FarEndCurrentEntry 4}
dsx3FarEndCurrentCSESs	::= {dsx3FarEndCurrentEntry 5}
dsx3FarEndCurrentCCVs	::= {dsx3FarEndCurrentEntry 6}
dsx3FarEndCurrentUASs	::= {dsx3FarEndCurrentEntry 7}
dsx3FarEndInvalidIntervals	::= {dsx3FarEndCurrentEntry 8}

The MIB objects are listed in Table 2-37.

Table 2-37 DS3/E3 Far End Current Table MIB Objects

Name	Syntax	Description	Default Value
dsx3FarEndCurrent Table	`Sequence of dsx3FarEndCurrentEntry`	Contains statistics being collected for the current 15-minute interval that are used for the DS3 Far End current table. The statistics are collected from the far end block error code within the C- bits. Max-Access: not-accessible Status: current	none
dsx3FarEndCurrent Entry	`dsx3FarEndCurrentEntry`	Provides an entry in the DS3 Far End Current table. Max-Access: not-accessible Status: current The index is `dsx3FarEndCurrentIndex`.	none
dsx3FarEndCurrent Index	`InterfaceIndex`	Specifies the index value which uniquely identifies the DS3 interface to which this entry applies. The interface identified by a particular value of this index is identical to the interface identified by the same value of `dsx3LineIndex`. Max-Access: read-only Status: current	none
dsx3FarEndTime Elapsed	`INTEGER (0..899)`	Determines the number of seconds that have elapsed since the beginning of the far end current error-measurement period. If, for some reason, the current interval exceeds the maximum value, the agent returns the maximum value. Max-Access: read-only Status: current	none
dsx3FarEndValid Intervals	`INTEGER (0..96)`	Determines the number of previous far end intervals for which data was collected. The value is 96 unless the interface was brought online within the last 24-hours. Then, the value will be the number of complete 15-minute far end intervals since the interface has been online. Max-Access: read-only Status: current	none
dsx3FarEndCurrent CESs	`PerfCurrentCount`	Describes the counter associated with the number of Far End C-bit Errored Seconds. Max-Access: read-only Status: current	none
dsx3FarEndCurrent CSESs	`PerfCurrentCount`	Describes the counter associated with the number of Far End C-bit Severely Errored Seconds. Max-Access: read-only Status: current	none

Table 2-37 DS3/E3 Far End Current Table MIB Objects (continued)

Name	Syntax	Description	Default Value
dsx3FarEndCurrent CCVs	`PerfCurrentCount`	Describes the counter associated with the number of Far End C-bit Coding Violations reported through the far end block error count. Max-Access: read-only Status: current	none
dsx3FarEndCurrent UASs	`PerfCurrentCount`	Describes the counter associated with the number of Far End unavailable seconds. Max-Access: read-only Status: current	none
dsx3FarEndInvalid Intervals	`INTEGER (0..96)`	Determines the number of intervals in the range from 0 to `dsx3FarEndValidIntervals`. No data is available. This object is generally 0 except in cases where the data is not available for some intervals, for example, in proxy situations. Max-Access: read-only Status: current	none

DS3 Far End Interval Table

The object identifiers are listed in Table 2-38.

Table 2-38 DS3/E3 Far End Interval Table Object Identifiers

Name	Object Identifier
dsx3FarEndIntervalTable	::= {ds3 11}
dsx3FarEndIntervalEntry	::= {dsx3FarEndIntervalTable 1}
dsx3FarEndIntervalIndex	::= {dsx3FarEndIntervalEntry 1}
dsx3FarEndIntervalNumber	::= {dsx3FarEndIntervalEntry 2}
dsx3FarEndIntervalCESs	::= {dsx3FarEndIntervalEntry 3}
dsx3FarEndIntervalCSESs	::= {dsx3FarEndIntervalEntry 4}
dsx3FarEndIntervalCCVs	::= {dsx3FarEndIntervalEntry 5}
dsx3FarEndIntervalUASs	::= {dsx3FarEndIntervalEntry 6}
dsx3FarEndIntervalValidData	::= {dsx3FarEndIntervalEntry 7}

The MIB objects are listed in Table 2-39.

Table 2-39 DS3/E3 Far End Interval Table MIB Objects

Name	Syntax	Description	Default Value
dsx3FarEndInterval Table	`Sequence of dsx3FarEndIntervalEntry`	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. Max-Access: not-accessible Status: current	none
dsx3FarEndInterval Entry	`dsx3FarEndIntervalEntry`	Provides an entry in the DS3 Far End Interval table. Max-Access: not-accessible Status: current The indexes include: • `dsx3FarEndIntervalIndex` • `dsx3FarEndIntervalNumber`	none
dsx3FarEndInterval Index	`InterfaceIndex`	Specifies the index value that identifies the DS3 interface to which this entry is applicable. The interface identified by a particular value of this index is identical to the interface identified by the same value of `dsx3LineIndex`. Max-Access: read-only Status: current	none
dsx3FarEndInterval Number	`INTEGER (1..96)`	Determines a number from 1 to 96. For example, 1 is the most recently completed 15-minute interval. 96 is the 15-minute interval completed for 23 hours and 45 minutes prior to interval 1. Max-Access: read-only Status: current	none
dsx3FarEndInterval CESs	`PerfIntervalCount`	Describes the counter associated with the number of Far End C-bit Errored Seconds. The counter is encountered by a DS3 interface in one of the previous 96, individual 15-minute intervals. In the case where the agent is a proxy and data is not available, return `noSuchInstance`. Max-Access: read-only Status: current	none
dsx3FarEndInterval CSESs	`PerfIntervalCount`	Describes the counter associated with the number of Far End C-bit Severely Errored Seconds. Max-Access: read-only Status: current	none
dsx3FarEndInterval CCVs	`PerfIntervalCount`	Describes the counter associated with the number of Far End C-bit Coding Violations reported through the far end block error count. Max-Access: read-only Status: current	none

Table 2-39 DS3/E3 Far End Interval Table MIB Objects (continued)

Name	Syntax	Description	Default Value
dsx3FarEndIntervalUASs	`PerfIntervalCount`	Describes the counter associated with the number of Far End unavailable seconds. Max-Access: read-only Status: current	none
dsx3FarEndIntervalValidData	`TruthValue`	Indicates if the data for this interval is valid. Max-Access: read-only Status: current	none

DS3 Far End Total Table

The object identifiers are listed in Table 2-40.

Table 2-40 DS3 Far End Total Table Object Identifiers

Name	Object Identifier
dsx3FarEndTotalTable	::= {ds3 12}
dsx3FarEndTotalEntry	::= {dsx3FarEndTotalTable 1}
dsx3FarEndTotalIndex	::= {dsx3FarEndTotalEntry 1}
dsx3FarEndTotalCESs	::= {dsx3FarEndTotalEntry 2}
dsx3FarEndTotalCSESs	::= {dsx3FarEndTotalEntry 3}
dsx3FarEndTotalCCVs	::= {dsx3FarEndTotalEntry 4}
dsx3FarEndTotalUASs	::= {dsx3FarEndTotalEntry 5}

The MIB objects are listed in Table 2-41.

Table 2-41 DS3 Far End Total Table MIB Objects

Name	Syntax	Description	Default Value
dsx3FarEndTotalTable	`Sequence of dsx3FarEndTotalEntry`	Contains the cumulative sum of the various statistics for the 24-hour period preceding the current interval for the DS3 Far End Total Table. Max-Access: not-accessible Status: current	none
dsx3FarEndTotalEntry	`dsx3FarEndTotalEntry`	Provides an entry in the DS3 Far End Total table. Max-Access: not-accessible Status: current The index is `dsx3FarEndTotalIndex`.	none

Table 2-41 DS3 Far End Total Table MIB Objects (continued)

Name	Syntax	Description	Default Value
dsx3FarEndTotal Index	`InterfaceIndex`	Specifies the index value which uniquely identifies the DS3 interface to which this entry applies. The interface identified by a particular value of this index is identical to the interface identified by the same value of `dsx3LineIndex`. Max-Access: read-only Status: current	none
dsx3FarEndTotal CESs	`PerfTotalCount`	Describes the counter associated with the number of Far End C-bit Errored Seconds encountered by a DS3 interface in the previous 24-hour interval. Invalid 15-minute intervals count as 0. Max-Access: read-only Status: current	none
dsx3FarEndTotal CSESs	`PerfTotalCount`	Describes the counter associated with the number of Far End C-bit Severely Errored Seconds encountered by a DS3 interface in the previous 24-hour interval. Invalid 15-minute intervals count as 0. Max-Access: read-only Status: current	none
dsx3FarEndTotal CCVs	`PerfTotalCount`	Describes the counter associated with the number of Far End C-bit Coding Violations. It is reported through the far end block error count, which is encountered by a DS3 interface in the previous 24-hour interval. Invalid 15-minute intervals count as 0. Max-Access: read-only Status: current	none
dsx3FarEndTotal UASs	`PerfTotalCount`	Describes the counter associated with the number of Far End unavailable seconds encountered by a DS3 interface in the previous 24-hour interval. Invalid 15-minute intervals count as 0. Max-Access: read-only Status: current	none

ds3 Traps

The object identifiers are listed in Table 2-42.

Table 2-42 ds3 Traps Object Identifiers

Name	Object Identifier
ds3Traps	::= {ds3 15}
dsx3LineStatusChange	::= {ds3Traps 0 1}

One MIB object is listed in Table 2-43.

Table 2-43 ds3 Traps MIB Object

Name	Objects	Description	Default Value
dsx3LineStatus Change	`dsx3LineStatus, dsx3LineStatusLastChange`	Specifies that a `dsx3LineStatusChange` trap is sent when the value of an instance changes. An NMS uses it to trigger polls. When the line status change results in a lower level line status change, for example, DS1, no traps for the lower level are sent. Status: current	none

DS/E3 Conformance Information

The object identifiers are listed in Table 2-44.

Table 2-44 DS/E3 Conformance Object Identifiers

Name	Object Identifier
ds3Conformance	::= {ds3 14}
ds3Groups	::= {ds3Conformance 1}
ds3Compliances	::= {ds3Conformance 2}
dsx3Channelization	::= {ds3Compliances 1}

The MIB objects are listed in Table 2-45.

Table 2-45 DS/E3 Conformance MIB Objects

Name	Mandatory Groups	Description	Default Value
ds3Compliance	ds3NearEndConfig Group, ds3NearEndStatistics Group	Describes the compliance statement for the DS3/E3 interfaces. Status: current Module: AXSM The mandatory groups include: **Group** — **Description** ds3FarEndGroup — Implementation of this group is optional for all systems that attach to a DS3 Interface. However, only C-bit Parity and SYNTRAN DS3 applications can provide this information. ds3NearEndOptional ConfigGroup — Implementation of this group is optional for all systems that attach to a DS3 interface.	none
dsx3LineType	none	Specifies that write access for the line type is not required. Min-Access: read-only	none
dsx3LineCoding	none	Specifies that write access for the line coding is not required. Min-Access: read-only	none
dsx3SendCode	none	Specifies that write access for the send code is not required. Min-Access: read-only	none
dsx3Loopback Config	none	Specifies that write access for loopbacks is not required. Min-Access: read-only	none
dsx3Transmit ClockSource	none	Specifies that write access for the transmit clock source is not required. Min-Access: read-only	none
dsx3LineLength	none	Specifies that write access for the line length is not required. Min-Access: read-only	none
dsx3Channelization	none	Specifies that write access for the channelization is not required. Min-Access: read-only	none

DS/E3 Units of Conformance

The object identifiers are listed in Table 2-46.

Table 2-46 DS/E3 Units of Conformance Object Identifiers

Name	Object Identifier
ds3NearEndConfigGroup	::= {ds3Groups 1}
ds3NearEndStatisticsGroup	::= {ds3Groups 2}
ds3FarEndGroup	::= {ds3Groups 3}
ds3DeprecatedGroup	::= {ds3Groups 4}
ds3NearEndOptionalConfigGroup	::= {ds3Groups 5}
ds3NearEndOptionalTrapGroup	::= {ds3Groups 6}

The MIB objects are listed in Table 2-47.

Table 2-47 DS/E3 Units of Conformance MIB Objects

Name	Objects	Description	Default Value
ds3NearEndConfig Group	dsx3LineIndex, dsx3TimeElapsed, dsx3ValidIntervals, dsx3LineType, dsx3LineCoding, dsx3SendCode, dsx3CircuitIdentifier, dsx3LoopbackConfig, dsx3LineStatus, dsx3TransmitClockSource, dsx3InvalidIntervals, dsx3LineLength, dsx3LoopbackStatus, dsx3Channelization, dsx3Ds1ForRemoteLoop	Describes a collection of objects providing configuration information applicable to all DS3/E3 interfaces. Status: current	none
ds3NearEndStatistics Group	dsx3CurrentIndex, dsx3CurrentPESs, dsx3CurrentPSESs, dsx3CurrentSEFSs, dsx3CurrentUASs, dsx3CurrentLCVs, dsx3CurrentPCVs, dsx3CurrentLESs, dsx3CurrentCCVs, dsx3CurrentCESs, dsx3CurrentCSESs, dsx3IntervalIndex, dsx3IntervalNumber, dsx3IntervalPESs, dsx3IntervalPSESs, dsx3IntervalSEFSs, dsx3IntervalUASs, dsx3IntervalLCVs, dsx3IntervalPCVs, dsx3IntervalLESs dsx3IntervalCCVs, dsx3IntervalCESs, dsx3IntervalCSESs, dsx3IntervalValidData, dsx3TotalIndex, dsx3TotalPESs, dsx3TotalPSESs, dsx3TotalSEFSs, dsx3TotalUASs, dsx3TotalLCVs, dsx3TotalPCVs, dsx3TotalLESs, dsx3TotalCCVs, dsx3TotalCESs, dsx3TotalCSESs	Describes a collection of objects providing statistics information applicable to all DS3/E3 interfaces. Status: current	none

Table 2-47 DS/E3 Units of Conformance MIB Objects (continued)

Name	Objects	Description	Default Value
ds3FarEndGroup	dsx3FarEndLineIndex, dsx3FarEndEquipCode, dsx3FarEndLocationIDCode, dsx3FarEndFrameIDCode, dsx3FarEndUnitCode, dsx3FarEndFacilityIDCode, dsx3FarEndCurrentIndex, dsx3FarEndTimeElapsed, dsx3FarEndValidIntervals, dsx3FarEndCurrentCESs, dsx3FarEndCurrentCSESs, dsx3FarEndCurrentCCVs, dsx3FarEndCurrentUASs, dsx3FarEndInvalidIntervals, dsx3FarEndIntervalIndex, dsx3FarEndIntervalNumber, dsx3FarEndIntervalCESs, dsx3FarEndIntervalCSESs, dsx3FarEndIntervalCCVs, dsx3FarEndIntervalUASs, dsx3FarEndIntervalValidData, dsx3FarEndTotalIndex, dsx3FarEndTotalCESs, dsx3FarEndTotalCSESs, dsx3FarEndTotalCCVs, dsx3FarEndTotalUASs	Describes a collection of objects providing remote configuration and statistics information applicable to C-bit Parity and SYNTRAN DS3 interfaces. Status: current	none
ds3DeprecatedGroup	dsx3IfIndex, dsx3FracIndex, dsx3FracNumber, dsx3FracIfIndex	Describes a collection of obsolete objects implemented for backwards compatibility. Status: deprecated	none
ds3NearEndOptional ConfigGroup	dsx3LineStatusChangeTrap Enable	Describes a collection of objects implemented on DS3/E3 interfaces. Status: current	none
ds3NearEndOptional TrapGroup	none	Describes a collection of notifications implemented on DS3/E3 interfaces. Notifications: dsx3LineStatusChange Status: current	none

Internet Assigned Numbers Authority Interface Textual Convention

This section describes a single MIB object that resides in the IANAifType-MIB.my file.

Note The IANAifType MIB is supported by AXSM and PXM45.

One Internet Assigned Numbers Authority (IANA) Interface textual convention is listed in Table 2-48.

Table 2-48 IANA Textual Convention MIB Object

Name	Syntax	Description	Default Value
IANAifType	INTEGER { other(1), -- none of the following regular1822(2),hdh1822(3), ddnX25(4),rfc877x25(5), ethernetCsmacd(6),iso88023Csmacd(7), iso88024TokenBus(8),iso88025TokenRing(9), iso88026Man(10),starLan(11), proteon10Mbit(12),proteon80Mbit(13), hyperchannel(14),fddi(15), lapb(16),sdlc(17), ds1(18), -- DS1-MIB e1(19), -- Obsolete see DS1-MIB basicISDN(20),primaryISDN(21), propPointToPointSerial(22), -- proprietary serial ppp(23), softwareLoopback(24), eon(25), -- CLNP over IP ethernet3Mbit(26), nsip(27), -- XNS over IP slip(28), -- generic SLIP ultra(29), -- ULTRA technologies ds3(30), -- DS3-MIB sip(31), -- SMDS, coffee frameRelay(32), -- DTE only. rs232(33), para(34), -- parallel-port arcnet(35), -- arcnet arcnetPlus(36), -- arcnet plus atm(37), -- ATM cells miox25(38), sonet(39), -- SONET or SDH x25ple(40),iso88022llc(41), localTalk(42), smdsDxi(43),	Specifies the syntax of the ifType object in the current definition of the MIB-II ifTable. The definition of this textual convention with the addition of newly assigned values is published periodically by the IANA These definitions and values can be obtained either by the Assigned Numbers RFC or specific Internet Network Management number assignments. The latest arrangements can be obtained by contacting the IANA. Requests for new values can be made to IANA through email iana@iana.org. The relationship between the assignment of ifType values and OIDs to particular media-specific MIBs are done by the IANA and is subject to change without notice. Quite often, a media-specific MIB OID-subtree assignment within the MIB-II transmission subtree is the same as the ifType value. However, in some circumstances, implementors must not presume any specific relationship between ifType values and the transmission subtree OID. Status: current	none

Table 2-48 IANA Textual Convention MIB Object (continued)

Name	Syntax	Description	Default Value
IANAifType (continued)	frameRelayService(44),-- FRNETSERV-MIB v35(45), hssi(46), hippi(47) modem(48), -- Generic modem aal5(49), -- AAL5 over ATM sonetPath(50),sonetVT(51), smdsIcip(52), -- SMDS InterCarrier Interface propVirtual(53), -- proprietary virtual/internal propMultiplexor(54),-- proprietary multiplexing ieee80212(55), -- 100BaseVG fibreChannel(56), -- Fibre Channel hippiInterface(57), -- HIPPI interfaces frameRelayInterconnect(58), -- Obsolete use either -- frameRelay(32) or -- frameRelayService(44). aflane8023(59), -- ATM Emulated LAN for 802.3 aflane8025(60), -- ATM Emulated LAN for 802.5 cctEmul(61), -- ATM Emulated circuit fastEther(62), -- Fast Ethernet (100BaseT) isdn(63), -- ISDN and X.25 v11(64), -- CCITT V.11/X.21 v36(65), -- CCITT V.36 g703at64k(66), -- CCITT G703 at 64Kbps g703at2mb(67), -- Obsolete see DS1-MIB qllc(68), -- SNA QLLC fastEtherFX(69), -- Fast Ethernet (100BaseFX) channel(70), -- channel ieee80211(71), -- radio spread spectrum ibm370parChan(72), -- IBM System 360/370 OEMI Channel escon(73), -- IBM Enterprise Systems Connection dlsw(74), -- Data Link Switching isdns(75), -- ISDN S/T interface isdnu(76), -- ISDN U interface lapd(77), -- Link Access Protocol D ipSwitch(78), -- IP Switching Objects rsrb(79), -- Remote Source Route Bridging atmLogical(80), -- ATM Logical Port ds0(81), -- Digital Signal Level 0 ds0Bundle(82), -- group of ds0s on the same ds1 bsc(83), -- Bisynchronous Protocol async(84), -- Asynchronous Protocol cnr(85), -- Combat Net Radio iso88025Dtr(86), -- ISO 802.5r DTR		

Table 2-48 IANA Textual Convention MIB Object (continued)

Name	Syntax	Description	Default Value
IANAifType (continued)	eplrs(87), -- Ext Pos Loc Report Sys arap(88), -- Appletalk Remote Access Protocol propCnls(89), -- Proprietary Connectionless Protocol hostPad(90), -- CCITT-ITU X.29 PAD Protocol termPad(91), -- CCITT-ITU X.3 PAD Facility frameRelayMPI(92), -- Multiproto Interconnect over FR x213(93), -- CCITT-ITU X213 adsl(94), -- Asymmetric Digital Subscriber Loop radsl(95), -- Rate-Adapt. Digital Subscriber Loop sdsl(96), -- Symmetric Digital Subscriber Loop vdsl(97), -- Very H-Speed Digital Subscrib. Loop iso88025CRFPInt(98), -- ISO 802.5 CRFP myrinet(99), -- Myricom Myrinet voiceEM(100), -- voice recEive and transMit voiceFXO(101), -- voice Foreign Exchange Office voiceFXS(102), -- voice Foreign Exchange Station voiceEncap(103), -- voice encapsulation voiceOverIp(104), -- voice over IP encapsulation atmDxi(105), -- ATM DXI atmFuni(106), -- ATM FUNI atmIma (107), -- ATM IMA pppMultilinkBundle(108), -- PPP Multilink Bundle ipOverCdlc (109), -- IBM ipOverCdlc ipOverClaw (110), -- IBM Common Link Access to Workstn stackToStack (111), -- IBM stackToStack virtualIpAddress (112), -- IBM VIPA mpc (113), -- IBM multi-protocol channel support ipOverAtm (114), -- IBM ipOverAtm iso88025Fiber (115), -- ISO 802.5j Fiber Token Ring tdlc (116), -- IBM twinaxial data link control gigabitEthernet (117), -- Gigabit Ethernet hdlc (118), -- HDLC, lapf (119), -- LAP F v37 (120), -- V.37 x25mlp (121), -- Multi-Link Protocol x25huntGroup (122), -- X25 Hunt Group trasnpHdlc (123), -- Transp HDLC		

Table 2-48 IANA Textual Convention MIB Object (continued)

Name	Syntax	Description	Default Value
IANAifType (continued)	interleave (124), -- Interleave channel fast (125), -- Fast channel ip (126), -- IP (for APPN HPR in IP networks) docsCableMaclayer (127), -- CATV Mac Layer docsCableDownstream (128), -- CATV Downstream interface docsCableUpstream (129), -- CATV Upstream interface a12MppSwitch (130), -- Avalon Parallel Processor tunnel (131), -- Encapsulation interface coffee (132), -- coffee pot ces (133), -- Circuit Emulation Service atmSubInterface (134), -- ATM Sub Interface l2vlan (135), -- Layer 2 Virtual LAN using 802.1Q l3ipvlan (136), -- Layer 3 Virtual LAN using IP l3ipxvlan (137), -- Layer 3 Virtual LAN using IPX digitalPowerline (138), -- IP over Power Lines mediaMailOverIp (139), -- Multimedia Mail over IP dtm (140), -- Dynamic syncronous Transfer Mode dcn (141), -- Data Communications Network ipForward (142), -- IP Forwarding Interface msdsl (143), -- Multi-rate Symmetric DSL ieee1394 (144), -- IEEE1394 High Performance Serial Bus if-gsn (145), -- HIPPI-6400 dvbRccMacLayer (146), -- DVB-RCC MAC Layer dvbRccDownstream (147), -- DVB-RCC Downstream Channel dvbRccUpstream (148), -- DVB-RCC Upstream Channel atmVirtual (149), -- ATM Virtual Interface mplsTunnel (150), -- MPLS Tunnel Virtual Interface srp (151),-- Spatial Reuse Protocol voiceOverAtm (152), -- Voice Over ATM voiceOverFrameRelay (153), -- Voice Over Frame Relay idsl (154), -- Digital Subscriber Loop over ISDN compositeLink (155), -- Avici Composite Link Interface ss7SigLink (156), -- SS7 Signaling Link propWirelessP2P (157), -- Prop. P2P wireless interface		

Table 2-48 IANA Textual Convention MIB Object (continued)

Name	Syntax	Description	Default Value
IANAifType (continued)	frForward (158), -- Frame Forward Interface rfc1483 (159), -- Multiprotocol over ATM AAL5 usb (160), -- USB Interface ieee8023adLag(161), -- IEEE 802.3ad Link Aggregate bgppolicyaccounting(162), -- BGP Policy Accounting frf16MfrBundle (163), -- FRF .16 Multilink Frame Relay h323Gatekeeper (164), -- H323 Gatekeeper h323Proxy (165), -- H323 Voice and Video Proxy mpls (166), -- MPLS mfSigLink (167), -- Multi-frequency signaling link hdsl2 (168), -- High Bit-Rate DSL - 2nd generation shdsl (169) -- Multirate HDSL2}		

Interfaces Group MIB Objects

This section describes the Interfaces Group MIB modules, which reside in the IF-MIB.my file. The MIB objects are extracted from RFC 2233 and describe the generic network interface sublayers. These MIB objects are an updated version of MIB-II's `ifTable`, and incorporate the extensions defined in RFC 1229.

Note The IF MIB is supported by AXSM, AXSM-E, PXM45, and RPM, which is also implemented in PXM45.

The Interfaces Group MIB include:

- Textual Conventions
- Interface Index
- Interfaces Table
- Extension to the Interface Table
- High Capacity Counters
- Generic Receive Address Table
- Interface-Related Traps
- Interfaces Group Conformance Information

Textual Conventions

The names of the textual conventions are specified in the object syntax.

The object identifiers are listed in Table 2-49.

Table 2-49 Interfaces Group Textual Convention Object Identifiers

Name	Object Identifier
ifNumber	::= {interfaces 1}
fTableLastChange	::= {ifMIBs 5}

The MIB objects are listed in Table 2-50.

Table 2-50 Interfaces Group Textual Convention MIB Objects

Name	Syntax	Description	Default Value
OwnerString[1]	`OCTET STRING (SIZE(0..255))`	Specifies the data type that is used to model an administratively assigned name of the owner of a resource. This information is taken from the NVT ASCII character set. It is suggested that this name contain one or more of the following: • ASCII form of the manager station's transport address. • Management station name (for example, domain name). • Network management personnel name, location, or phone number. In some cases the agent itself will be the owner of an entry. In these cases, this string can be set to a string starting with agent. Display Hint: `255a` Status: current	none
InterfaceIndex[1]	`Integer32 (1..2147483647)`	Contains the semantics of `ifIndex` and is used for any syntax defined on other MIB modules that need these semantics. Specifies a unique value, greater than 0, for each interface or interface sublayer in the managed system. It is recommended that values be assigned contiguously starting from 1. The value for each interface sublayer must remain constant at least from one reinitialization of the entity's network management system to the next reinitialization. Display Hint: `d` Status: current	none

Table 2-50 Interfaces Group Textual Convention MIB Objects (continued)

Name	Syntax	Description	Default Value
InterfaceIndexOrZero[1]	`Integer32 (0..2147483647)`	Specifies an extension of the InterfaceIndex convention. The latter defines a greater than 0 value used to identify an interface or interface sublayer in the managed system. This extension permits the additional value of 0. The value 0 is specific and must therefore be defined as part of the description. Examples of the use of 0 include situations where interface was unknown, or when none or all interfaces need to be referenced. Display Hint: `d` Status: current	none
ifNumber	`Integer32`	Determines the number of network interfaces (regardless of their current state) present on this system. Max-Access: read-only Status: current	none
ifTableLastChange	`TimeTicks`	Specifies the value of `sysUpTime` at the time of the last creation or deletion of an entry in the `ifTable`. If the number of entries has been unchanged since the last reinitialization of the local network management subsystem, this contains a 0 value. Max-Access: read-only Status: current	none

1. Object identifier is not supported.

Interface Index

The interface index contains the semantics of the `ifIndex` and is used for any objects defined on other mib modules that need these semantics.

The object identifier for each MIB object is listed in Table 2-51.

Table 2-51 Interface Index Object Identifiers

Name	Object Identifier
OwnerString	none
InterfaceIndex	none
InterfaceIndexOrZero	none
ifNumber	::= { interfaces 1 }
ifTableLastChange	::= { ifMIBObjects 5 }

The MIB objects are listed in Table 2-52.

Table 2-52 Interface Index MIB Objects

Name	Syntax	Description	Default Value
OwnerString	`OCTET STRING (SIZE(0..255))`	Uses a model of an administratively-assigned name of the resource owner for this data type. This information is taken from the NVT ASCII character set. It is suggested that this name contain one or more of the following: • ASCII form of the manager station's transport address. • Management station name (for example, domain name). • Network management personnel name, location, or phone number. In some cases, the agent itself is the owner of an entry. In these cases, start of the string can be the agent substring. `InterfaceIndex` contains the semantics of `ifIndex` and is used for any syntax defined on other MIB modules that need these semantics. Display Hint: `255a` Status: current	none
InterfaceIndex	`Integer32 (1..2147483647)`	Specifies a unique value, greater than 0, for each interface or interface sublayer in the managed system. It is recommended that values are assigned contiguously starting from 1. The value for each interface sublayer must remain constant at least from one reinitialization of the entity network management system to the next reinitialization. Display Hint: `d` Status: current	none
InterfaceIndexOrZero	`Integer32 (0..2147483647)`	Specifies an extension of the `InterfaceIndex` convention for this textual convention. The latter defines a greater than 0 value used to identify an interface or interface sublayer in the managed system. This extension permits the additional value of 0. The 0 value is specific and must therefore be defined as part of the description of any object using this syntax. Examples of the use of 0 include situations when the interface was unknown, or when none or all interfaces are referenced. Display Hint: `d` Status: current	none

Table 2-52 Interface Index MIB Objects (continued)

Name	Syntax	Description	Default Value
ifNumber	`Integer32`	Determines the number of network interfaces (regardless of their current state) present on this system. Max-Access: read-only Status: current	none
ifTableLastChange	`TimeTicks`	Determines the value `sysUpTime` at the last creation or deletion of an entry in the `ifTable`. If the number of entries are unchanged since the last reinitialization of the local network management subsystem, this contains a 0 value. Max-Access: read-only Status: current	none

Interfaces Table

The Interfaces Table contains information on the entity interfaces. Each sublayer below the internetwork layer of a network interface is considered to be an interface.

The object identifiers are listed in Table 2-53.

Table 2-53 Interfaces Table Object Identifiers

Name	Object Identifier
ifTable	::= {interfaces 2}
ifEntry	::= {ifTable 1}
ifIndex	::= {ifEntry 1}
ifDescr	::= {ifEntry 2}
ifType	::= {ifEntry 3}
ifMtu	::= {ifEntry 4}
ifSpeed	::= { ifEntry 5 }
ifPhysAddress	::= {ifEntry 6}
ifAdminStatus	::= {ifEntry 7}
ifOperStatus	::= {ifEntry 8}
ifLastChange	::= {ifEntry 9}
ifInOctets	::= {ifEntry 10}
fInUcastPkts	::= {ifEntry 11}
ifInNUcastPkts	::= {ifEntry 12}
ifInDiscards	::= {ifEntry 13}
ifInErrors	::= {ifEntry 14}

Table 2-53 Interfaces Table Object Identifiers (continued)

Name	Object Identifier
ifInUnknownProtos	::= {ifEntry 15}
ifOutOctets	::= {ifEntry 16}
ifOutUcastPkts	::= {ifEntry 17}
ifOutNUcastPkts	::= {ifEntry 18}
ifOutDiscards	::= {ifEntry 19}
ifOutErrors	::= {ifEntry 20}
ifOutQLen	::= {ifEntry 21}
ifSpecific	::= {ifEntry 22}

The MIB objects are listed in Table 2-54.

Table 2-54 Interfaces Table MIB Objects

Name	Syntax	Description	Default Value
ifTable	`Sequence of IfEntry`	Lists the interface entries. The number of entries is given by the value of `ifNumber`. Max-Access: not-accessible Status: current	none
ifEntry	`ifEntry`	Provides an entry containing management information applicable to a particular interface. Max-Access: not-accessible Status: current The index is `ifIndex`.	none
ifIndex	`InterfaceIndex`	Specifies a unique value, greater than 0, for each interface. It is recommended that values are assigned starting with 1. The value for each interface sublayer must remain constant at least from one reinitialization of the entity's network management system to the next reinitialization. Max-Access: read-only Status: current	none
ifDescr	`DisplayString (SIZE (0..255))`	Specifies a textual string containing information about the interface. This string includes the name of the manufacturer, the product name, and the version of the interface hardware or software. Max-Access: read-only Status: current	none

Table 2-54 Interfaces Table MIB Objects (continued)

Name	Syntax	Description	Default Value
ifType	`IANAifType`	Describes the type of interface. Additional values for `ifType` are assigned by the Internet Assigned Numbers Authority (IANA), through updating the Syntax of the `IANAifType` textual convention. Max-Access: read-only Status: current	none
ifMtu	`Integer32`	Determines the size of the largest packet, which is sent/received on the interface. It is specified in octets. For interfaces that are used for transmitting network datagrams, this is the size of the largest network datagram that is sent on the interface. Max-Access: read-only Status: current	none
ifSpeed	`Gauge32`	Determines the estimate of the interface current bandwidth in bps. For interfaces which do not vary in bandwidth, or for those where no accurate estimation can be made, this can contain the nominal bandwidth. If the bandwidth of the interface is greater than the maximum value, the maximum value (4,294,967,295) and `ifHighSpeed` must be used to report the interface speed. For a sublayer which has no bandwidth, the value is 0. Max-Access: read-only Status: current	none
ifPhysAddress	`PhysAddress`	Specifies the interface address at its protocol sublayer. For example, an 802.x interface contains a MAC address. The interface media-specific MIB must define the bit and byte ordering and the format of the value of this address. For interfaces which do not have such an address, for example, a serial line, can contain an octet string of 0 length. Max-Access: read-only Status: current	none
ifAdminStatus	`Integer {up (1), ready to pass packets down; (2), testing; (3) in some test mode}`	Describes the desired state of the interface. The `testing(3)` state indicates that no operational packets are passed. When a managed system initializes, all interfaces start with `ifAdminStatus` in the `down(2)` state. As a result of either explicit management action or through configuration information retained by the managed system, `ifAdminStatus` is changed to either the `up(1)`, `testing(3)`, or remains in the `down(2)` state. Max-Access: read-write Status: current	none

Table 2-54 Interfaces Table MIB Objects (continued)

Name	Syntax	Description	Default Value
ifOperStatus	Integer {up (1), ready to pass packets; down(2); testing (3), in some test mode; unknown (4), status can not be determined; dormant (5); notPresent (6), some component is missing; lowerLayerDown (7), down due to state of lower-layer interface(s)}	Describes the current operational state of the interface. The testing (3) state indicates that no operational packets can be passed. If ifAdminStatus is changed to down (2), ifOperStatus changes to down(2). If the ifAdminStatus object is changed to up (1), ifOperStatus changes to up(1). If the interface is ready to transmit and receive network traffic; it can change to dormant(5) if the interface is waiting for external actions (such as a serial line waiting for an incoming connection). It remains in the down(2) state if and only if there is a fault that prevents it from going to the up(1) state. It remains in the notPresent (6) state if the interface has missing components typically hardware. Max-Access: read-only Status: current	none
ifLastChange	TimeTicks	Specifies the value of sysUpTime when the interface entered its current operational state. If the current state was entered before the last reinitialization of the local network management subsystem, this contains a 0 value. Max-Access: read-only Status: current	none
ifInOctets	Counter32	Determines the total number of octets received on the interface, including framing characters. Discontinuities in the value of this counter can occur at reinitialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime. Max-Access: read-only Status: current	none
ifInUcastPkts	Counter32	Determines the number of packets delivered by this sublayer to a higher sublayer, which are not addressed to a multicast or broadcast address at this sublayer. Discontinuities in the value of this counter can occur at reinitialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime. Max-Access: read-only Status: current	none
ifInNUcastPkts	Counter32	Determines the number of packets delivered by this sublayer to a sublayer, which are addressed to a multicast or broadcast address at this sublayer. Discontinuities in the value of this counter can occur at reinitialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime. This is discontinued in favor of ifInMulticastPkts and ifInBroadcastPkts. Max-Access: read-only Status: deprecated	none

Table 2-54 Interfaces Table MIB Objects (continued)

Name	Syntax	Description	Default Value
ifInDiscards	Counter32	Determines the number of inbound packets that were chosen to be discarded (even though no errors had been detected) to prevent their being deliverable to a higher layer protocol. One possible reason for discarding such a packet could be to free buffer space. Discontinuities in the value of this counter can occur at reinitialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime. Max-Access: read-only Status: current	none
ifInErrors	Counter32	Determines the number of inbound packets that contained errors preventing them from being delivered to a higher layer protocol for packet-oriented interfaces. For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher layer protocol. Discontinuities in the value of this counter can occur at reinitialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime. Max-Access: read-only Status: current	none
ifInUnknown Protos	Counter32	Determines the number of packets received through the interface that were discarded because of an unknown or unsupported protocol for packet-oriented interfaces. For character-oriented or fixed-length interfaces that support protocol multiplexing, the number of transmission units received through the interface that were discarded because of an unknown or unsupported protocol. For any interface that does not support protocol multiplexing, this counter is always 0. Discontinuities in the value of this counter can occur at reinitialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime. Max-Access: read-only Status: current	none
ifOutOctets	Counter32	Determines the total number of octets transmitted out of the interface, including framing characters. Discontinuities in the value of this counter can occur at reinitialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime. Max-Access: read-only Status: current	none

Table 2-54 Interfaces Table MIB Objects (continued)

Name	Syntax	Description	Default Value
ifOutUcastPkts	Counter32	Determines the total number of packets that higher-level protocols requested be transmitted, and which were not addressed to a multicast or broadcast address at this sublayer, including those that were discarded or not sent. Discontinuities in the value of this counter can occur at reinitialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime. Max-Access: read-only Status: current	none
ifOutNUcast Pkts	Counter32	Determines the total number of packets that higher-level protocols requested be transmitted, and that were addressed to a multicast or broadcast address at this sublayer, including those that were discarded or not sent. Discontinuities in the value of this counter can occur at reinitialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime. This is discontinued in favor of ifOutMulticastPkts and ifOutBroadcastPkts. Max-Access: read-only Status: deprecated	none
ifOutDiscards	Counter32	Determines the number of outbound packets that were chosen to be discarded (even though no errors had been detected) to prevent their being transmitted. One possible reason for discarding such a packet is to free buffer space. Discontinuities in the value of this counter can occur at reinitialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime. Max-Access: read-only Status: current	none
ifOutErrors	Counter32	Determines the number of outbound packets that cannot be transmitted because of errors for packet-oriented interfaces. For character-oriented or fixed-length interfaces, the number of outbound transmission units that can not be transmitted because of errors. Discontinuities in the value of this counter can occur at reinitialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime. Max-Access: read-only Status: current	none

Table 2-54 Interfaces Table MIB Objects (continued)

Name	Syntax	Description	Default Value
ifOutQLen	`Gauge32`	Determines the length of the output packet queue (in packets). Max-Access: read-only Status: deprecated	none
ifSpecific	`IDENTIFIER`	Points to an instance of a MIB in the media-specific MIB, for example, the semantics associated with the Instance Pointer textual convention defined in RFC 1903. It is recommended that the media-specific MIB specify what value `ifSpecific` can take for values of `ifType`. If no MIB definitions specific to the particular media are available, the value is set to the `IDENTIFIER {0 0}`. Max-Access: read-only Status: deprecated	none

Extension to the Interface Table

The object identifiers are listed in Table 2-55.

Note This table replaces the `ifExtnsTable` table.

Table 2-55 Extension to the Interface Table Object Identifiers

Name	Object Identifier
ifXTable	::= { ifMIBs 1 }
ifXEntry	::= { ifXTable 1 }
ifName	::= { ifXEntry 1 }
ifInMulticastPkts	::= { ifXEntry 2 }
ifInBroadcastPkts	::= { ifXEntry 3 }
ifOutMulticastPkts	::= { ifXEntry 4 }
ifOutBroadcastPkts	::= { ifXEntry 5 }

The MIB objects are listed in Table 2-56.

Table 2-56 Extension to the Interface Table MIB Objects

Name	Syntax	Description	Default Value
ifXTable	`SEQUENCE OF IfXEntry`	Lists the interface entries. The number of entries is given by the value of `ifNumber`. This table contains additional entries for the interface table. Max-Access: not-accessible Status: current	none
ifXEntry	`IfXEntry`	Contains additional management information applicable to a particular interface for this entry. Max-Access: not-accessible Status: current The index contains `ifEntry`.	none

Table 2-56 Extension to the Interface Table MIB Objects (continued)

<table>
<tr><th>Name</th><th>Syntax</th><th>Description</th><th>Default Value</th></tr>
<tr><td>ifName</td><td>DisplayString</td><td>Specifies the textual name of the interface. The ifName value is the name of the interface as assigned by the local device and is suitable for use in commands entered at the device console. This is a text name, such as le0, or a simple port number, such as 1, depending on the interface naming syntax of the device. If several entries in the ifTable together represent a single interface as named by the device, each has the same value of ifName.

Note For an agent that responds to SNMP queries concerning an interface on some other (proxy) device, the ifName value for such an interface is the proxy device local name.

If there is no local name, or if ifName is not applicable, this object contains a 0 length string.

Note For the Cisco MGX 8850, ifName contains the physical or logical association of a name.

Max-Access: read-only

Status: current

The actual values depend on the implementation. The values for the ifName object include:
<table>
<tr><th>ifDescr</th><th>ifName</th><th>Remark</th></tr>
<tr><td>sonet4.1.1</td><td>sonet.4.1.1</td><td>Association for SONET Line:
Logical Slot 4, Bay 1, Line Number 1</td></tr>
<tr><td>ds3.2.2.2</td><td>ds3.2.2.2</td><td>Association for DS3 Line:
Logical Slot 2, Bay 2, Line Number 2</td></tr>
<tr><td>atm.4.1.1.1</td><td>atm.4.1.1.1</td><td>Association of ATM Physical Interface:
Logical Slot 4, Bay 1, Line Number 1,
AtmPhy Number 1</td></tr>
<tr><td>atmVirtual.5.1.2.20.200</td><td>atmVirtual.5.1.2.20.200</td><td>Association of ATM Virtual Interface for Virtual Trunk:
Logical Slot 5, Bay 1, Line Number 2,
Virtual Interface 20, VPI Number 200</td></tr>
</table>
Physical lines, ATM physical lines, and ATM virtual interfaces are the three conventions used for the value of ifName objects.

Physical Lines

This format is used for Physical Lines such as SONET, DS3, DS1, and so on forth. The following is an example:

<lineType>.<logical-slot>.<bay>.<lineNumber></td><td>none</td></tr>
</table>

Table 2-56 Extension to the Interface Table MIB Objects (continued)

Name	Syntax	Description		Default Value
ifName (continued)		The values for the physical lines include:		
		`lineType`	Refers to SONET, DS3, DS1, Ethernet, and RS232.	
		`logical-slot`	Determines the logical slot number of the module.	
		`bay`	Determines the back card number.	
		`lineNumber`	Specifies the physical line number.	

		The following is a SONET physical line example:
		`sonet.4.1.1 => sonet, Logical Slot 4, Bay 1 , Physical Line 1`
		
		The following is a ds3.6.2.3 physical line example:
		`ds3.6.2.3 => ds3, Logical Slot 6, Bay 2, Physical Line 3`
		
		ATM Physical Interfaces
		The following are the format used for ATM Physical Interfaces:
		`atm.<logical-slot>.<bay>.<lineNumber>`
		
		The following is the values for the ATM Physical Interfaces:		

		`logical-slot`	Determines the logical slot number of the module.	
		`bay`	Determines the back card number.	
		`lineNumber`	Determines the physical line number.	

		The following are examples:
		`atm.4.2.2.1 => atm, Logical Slot 4, Bay 2, Physical Line 2, ATM Phy 1`
		`atm.4.1.2.2 => atm, Logical Slot 4, Bay 1, IMA Id 2, ATM Phy 2`
		
		ATM Virtual Interfaces
		The following format is used for ATM Virtual Interfaces (`atmVirtual`):
		`atmVirtual.<logical-slot>.<bay>.<lineNumber>.<vi-number>.[<vpi-number>]`
		
		The values for the ATM Virtual Interface include:		

		`logical-slot`	Determines the logical slot number of the module.	
		`bay`	Determines the back card number.	
		`lineNumber/Ima-Id`	Specifies the physical line number or Ima ID.	
		`vi-number`	Determines the virtual interface number.	

Table 2-56 Extension to the Interface Table MIB Objects (continued)

Name	Syntax	Description		Default Value
ifName (continued)		vpi-number	Determines the VPI number and exists only for virtual trunk.	

		The following is an example of the first value, above:
		`atmVirtual.3.1.2.33 => atmVirtual, Logical Slot 3, Bay 1 , Physical Line 2, VI 33`
		The following is an example of the second value:
		`atmVirtual.3.1.3.23 => atmVirtual, Logical Slot 3, Bay 1, IMA Id 3, VI 23`
		The following is an example of the third value:
		`atmVirtual.4.2.2.34.2000 => atmVirtual, Logical Slot 4, Bay 2, Physical Line 2, VI 34, VPI 2000`		

ifInMulticast Pkts	Counter32	Determines the number of packets delivered by this sublayer to a higher sublayer, which is addressed to a multicast address at this sublayer. For a MAC layer protocol, this includes both groups and functional addresses. The discontinued value of this counter can occur at reinitialization of the management system, and at other times, as indicated by the `ifCounterDiscontinuityTime` value. Max-Access: read-only Status: current		none
ifInBroadcast Pkts	Counter32	Determines the number of packets that are delivered by this sublayer to a higher sublayer that are addressed to a broadcast address. The discontinued value of this counter can occur at reinitialization of the management system, and at other times, as indicated by the `ifCounterDiscontinuityTime` value. Max-Access: read-only Status: current		none
ifOutMulticast Pkts	Counter32	Determines the number of packets that higher-level protocols requested be transmitted, which are addressed to a multicast address, including those that are discarded or not sent. Both group and functional addresses are included if requested by a MAC layer protocol. The discontinued value of this counter can occur at reinitialization of the management system, and at other times, as indicated by the `ifCounterDiscontinuityTime` value. Max-Access: read-only Status: current		none
ifOutBroadcast Pkts	Counter32	Determines the number of packets that higher-level protocols requested be transmitted, and that are addressed to a broadcast address at this sublayer. The object includes those that are discarded or not sent. The discontinued value of this counter can occur at reinitialization of the management system, and at other times, as indicated by the `ifCounterDiscontinuityTime` value. Max-Access: read-only Status: current		none

High Capacity Counters

High capacity counters are all 64-bit versions of the basic `ifTable` counters. These counters all have the same basic semantics as their 32-bit counterparts; their syntax is extended to 64 bits.

The object identifiers are listed in Table 2-57.

Table 2-57 High Capacity Counter Object Identifiers

Name	Object Identifier
ifHCInOctets	::= {ifXEntry 6}
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}
ifConnectorPresent	::= {ifXEntry 17}
ifAlias	::= {ifXEntry 18}
ifConnectorPresent	::= {ifXEntry 19}

The MIB objects are listed in Table 2-58.

Table 2-58 High Capacity Counter MIB Objects

Name	Syntax	Description	Default Value
ifHCInOctets	Counter64	Determines the total number of octets received on the interface, including framing characters. This is a 64-bit version of ifInOctets. The discontinued value of this counter can occur at reinitialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime. Max-Access: read-only Status: current	none
ifHCInUcastPkts	Counter64	Determines the number of packets delivered by this sublayer to a higher sublayer that were not addressed to a multicast or broadcast address. This is a 64-bit version of ifInUcastPkts. The discontinued value of this counter can occur at reinitialization of the management system and at other times as indicated by the value of ifCounterDiscontinuityTime. Max-Access: read-only Status: current	none
ifHCInMulticastPkts	Counter64	Determines the number of packets delivered by this sublayer to a higher sublayer that were addressed to a multicast address. For a MAC layer protocol, this includes both group and functional addresses. This is a 64-bit version of ifInMulticastPkts. The discontinued value of this counter can occur at reinitialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime. Max-Access: read-only Status: current	none
ifHCInBroadcastPkts	Counter64	Determines the number of packets delivered by this sublayer to a higher sublayer that were addressed to a broadcast address. This is a 64-bit version of ifInBroadcastPkts. The discontinued value of this counter can occur at reinitialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime. Max-Access: read-only Status: current	none
ifHCOutOctets	Counter64	Determines the total number of octets transmitted out of the interface, which includes framing characters. This is a 64-bit version of ifOutOctets. The discontinued value of this counter can occur at reinitialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime. Max-Access: read-only Status: current	none

Table 2-58 High Capacity Counter MIB Objects (continued)

Name	Syntax	Description	Default Value
ifHCOutUcastPkts	`Counter64`	Determines the total number of packets that higher-level protocols requested be transmitted. The packets are not addressed to a multicast or broadcast address at this sublayer, including those that were discarded or not sent. This is a 64-bit version of `ifOutUcastPkts`. The discontinued value of this counter can occur at reinitialization of the management system, and at other times as indicated by the value of `ifCounterDiscontinuityTime`. Max-Access: read-only Status: current	none
ifHCOutMulticastPkts	`Counter64`	Determines the total number of packets that higher-level protocols requested be transmitted. The packets are addressed to a multicast address at this sublayer, including those that were discarded or not sent. For a MAC layer protocol, this includes both group and functional addresses. This is a 64-bit version of `ifOutMulticastPkts`. The discontinued value of this counter can occur at reinitialization of the management system, and at other times as indicated by the value of `ifCounterDiscontinuityTime`. Max-Access: read-only Status: current	none
ifHCOutBroadcastPkts	`Counter64`	Determines the total number of packets that higher-level protocols requested be transmitted. The packets are addressed to a broadcast address at this sublayer, including those that were discarded or not sent. This is a 64-bit version of `ifOutBroadcastPkts`. The discontinued value of this counter can occur at reinitialization of the management system, and at other times as indicated by the value of `ifCounterDiscontinuityTime`. Max-Access: read-only Status: current	none
ifLinkUpDownTrap Enable	`Integer {enabled (1), disabled (2)}`	Indicates whether `linkUp` or `linkDown` traps are generated for this interface. By default, this can have the value `enabled (1)` for interfaces which do not operate on top of any other interface (as defined in the `ifStackTable`), and `disabled (2)`. Max-Access: read-write Status: current	none

Table 2-58 High Capacity Counter MIB Objects (continued)

Name	Syntax	Description	Default Value
ifHighSpeed	`Gauge32`	Determines an estimate of the interface current bandwidth in units of 1,000,000 bps. If this reports a value of *n*, the speed of the interface is somewhere in the range of *n*-500,000 to *n*+499,999. For interfaces that do not vary in bandwidth or for which no accurate estimation is made, this contains the nominal bandwidth. For a sublayer that has a defined bandwidth, the value is 0. Max-Access: read-only Status: current	none
ifPromiscuousMode	`TruthValue`	Defines a value of `false (2)` if the interface accepts only packets and frames that are addressed to this station. It has a value of `true (1)` when the station accepts all packets and frames transmitted on the media. The value of `true (1)` is valid only on certain types of media. If valid, setting this to a value of `true (1)` requires the interface to be reset before becoming effective. The value of `ifPromiscuousMode` does not affect the reception of broadcast and multicast packets and frames by the interface. Max-Access: read-write Status: current	none
ifConnectorPresent	`TruthValue`	Defines the value `true (1)` if the interface sublayer has a physical connector and the value `false (2)`. Max-Access: read-only Status: current	none

Table 2-58 High Capacity Counter MIB Objects (continued)

Name	Syntax	Description	Default Value
ifAlias	`DisplayString (SIZE(0..64))`	Defines an alias for the interface as specified by a network manager, and provides a nonvolatile handle for the interface. On the first instance of an interface, the value of `ifAlias` associated with that interface is the zero-length string. When a value is written into an instance of `ifAlias` through a network management set operation, the agent must retain the supplied value in the `ifAlias` instance. It is associated with the same interface for as long as that interface remains instantiated. The following are included: • All reinitializations and reboots of the network management system. • Change of the interface `ifIndex` value. An example of the value that a network manager can store for a WAN interface is the Telco circuit number and identifier of the interface. Some agents can support write-access only for interfaces having particular values of `ifType`. An agent that supports write access is required to keep the value in nonvolatile storage. But it can limit the length of new values, depending on how much storage is already occupied by the values for other interfaces. Max-Access: read-write Status: current	none
ifCounterDiscontinuity Time	`TimeStamp`	Specifies the value of `sysUpTime` on the most recent occasion when any of the interface counters suffered a discontinuity. The relevant counters are the specific instances associated with this interface of any `Counter32` or `Counter64` contained in the `ifTable` or `ifXTable`. If no discontinued counter values occurred since the last reinitialization of the local management subsystem, this value is 0. Max-Access: read-only Status: current	none

Generic Receive Address Table

The Generic Receive Address Table group of objects is mandatory for all types of interfaces that can receive packets/frames addressed to more than one address.

Note This table replaces the `ifExtnsRcvAddr` table.

The main difference is that this table makes use of the `RowStatus` textual convention, while `ifExtnsRcvAddr` does not.

The object identifiers are listed in Table 2-59.

Table 2-59 Generic Receive Address Table Object Identifiers

Name	Object Identifier
ifRcvAddressTable	::= {ifMIBs 4}
ifRcvAddressEntry	::= {ifRcvAddressTable 1}
ifRcvAddressAddress	::= {ifRcvAddressEntry 1}
ifRcvAddressStatus	::= {ifRcvAddressEntry 2}
ifRcvAddressType	::= {ifRcvAddressEntry 3}

The MIB objects are listed in Table 2-60.

Table 2-60 Generic Receive Address Table MIB Objects

Name	Syntax	Description	Default Value
ifRcvAddressTable	`Sequence of ifRcvAddressEntry`	Contains an entry for each address (broadcast, multicast, or unicast) for which the system can receive packets/frames on a particular interface, except as follows: • For interfaces operating in promiscuous mode, entries are required only for those addresses for which the system receives frames. • For 802.5 functional addresses, only 1 entry is required. The address is the result of the functional address bit ANDed with the bit mask of all functional addresses for which the system receives frames. A system is normally able to use any unicast address that corresponds to an entry in this table as a source address. Max-Access: not-accessible Status: current	none
ifRcvAddressEntry	`ifRcvAddressEntry`	Identifies an address for which the system can accept packets/frames on the particular interface identified by the index value `ifIndex`. Max-Access: not-accessible Status: current The following are the indexes: • `ifIndex` • `ifRcvAddressAddress`	none
ifRcvAddressAddress	`PhysAddress`	Specifies an address for which the system can accept packets/frames on this entry interface. Max-Access: not-accessible Status: current	none

Table 2-60 Generic Receive Address Table MIB Objects (continued)

Name	Syntax	Description	Default Value
ifRcvAddressStatus	`RowStatus`	Creates and deletes rows in the `ifRcvAddressTable`. Max-Access: read-create Status: current	none
ifRcvAddressType	`Integer { other(1), volatile(2), nonVolatile(3)}`	Defines the value `nonVolatile (3)` for entries in the table that are valid and cannot be deleted by the next restart of the managed system. Entries having the value `volatile (2)` are valid and exist, but have not been saved, so these entries will not exist after the next restart of the managed system. Entries having the value `other (1)` are valid and exist, but are not classified as to whether they can continue to exist after the next restart. Max-Access: read-create Status: current	volatile

Interface-Related Traps

The object identifiers are listed in Table 2-61.

Table 2-61 Interface-Related Traps Object Identifiers

Name	Object Identifier
linkDown	::= {snmpTraps 3}
linkUp	::= {snmpTraps 4}

The MIB objects are listed in Table 2-62.

Table 2-62 Interface-Related Traps MIB Objects

Name	Syntax	Description	Default Value
linkDown	`{ifIndex, ifAdminStatus, ifOperStatus}`	Signifies that the SNMPv2 entity, acting in an agent role, has detected that the `ifOperStatus` for one of its communication links is about to enter the down state from some other state (but not from the `notPresent` state). This other state is indicated by the included value of `ifOperStatus`. Status: current	none
linkUp	`{ifIndex, ifAdminStatus, ifOperStatus}`	Signifies that the SNMPv2 entity, acting in an agent role, has detected that the `ifOperStatus` for one of its communication links left the down state and transitioned into some other state (but not into the `notPresent` state). This other state is indicated by the included value of `ifOperStatus`. Status: current	none

Interfaces Group Conformance Information

The object identifiers are listed in Table 2-63.

Note The four groups `ifPacketGroup`, `ifHCPacketGroup`, `ifVHCPacketGroup`, and `ifRcvAddressGroup` are mutually exclusive; only one of these groups is implemented for any interface.

Table 2-63 Interfaces Group Conformance Object Identifiers

Name	Object Identifier
ifCompliance2E	none
ifLinkUpDownTrapEnable	none
ifPromiscuousMode	none
ifStackStatus	none
ifAdminStatus	none
ifAlias	::= {ifCompliances 2}
ifGeneralInformationGroup	::= {ifGroups 10}
ifFixedLengthGroup	::= {ifGroups 2}
ifHCFixedLengthGroup	::= {ifGroups 3}
ifPacketGroup	::= {ifGroups 4}
ifHCPacketGroup	::= {ifGroups 5}
ifVHCPacketGroup	::= {ifGroups 6}
fRcvAddressGroup	::= {ifGroups 7}
ifStackGroup2	::= {ifGroups 11}
ifCounterDiscontinuityGroup	::= {ifGroups 13}

The MIB objects are shown in Table 2-64.

Table 2-64 Interfaces Group Conformance MIB Objects

<table>
<tr><th>Name</th><th>Syntax</th><th>Description</th><th>Default Value</th></tr>
<tr><td>ifCompliance2E</td><td>none</td><td>Describes the compliance statement for SNMPv2 entities that have network interfaces.
Status: current
Module: AXSM
The following are the mandatory groups:
ifGeneralInformationGroup
ifStackGroup2
ifCounterDiscontinuityGroup
The following are the descriptions for each group:<table><tr><th>Group</th><th>Description</th></tr><tr><td>ifFixedLength Group</td><td>This group is mandatory for all network interfaces which are character-oriented or transmit data in fixed-length transmission units.</td></tr><tr><td>ifHCFixedLength Group</td><td>This group is mandatory for only those network interfaces which are character-oriented or transmit data in fixed-length transmission units, and for which the value of the corresponding instance of ifSpeed is greater than 20 million bits/second.</td></tr><tr><td>ifPacketGroup</td><td>This group is mandatory for all network interfaces which are packet-oriented.</td></tr><tr><td>ifHCPacketGroup</td><td>This group is mandatory only for those network interfaces which are packet-oriented and for which the value of the corresponding instance of ifSpeed is greater than 650 million bits/second.</td></tr><tr><td>ifRcvAddress Group</td><td>The applicability of this group must be defined by the media-specific MIBs. Media-specific MIBs must define the exact meaning, use, and semantics of the addresses in this group.</td></tr></table></td><td>none</td></tr>
<tr><td>ifLinkUpDown TrapEnable</td><td>none</td><td>Specifies that write access is not required.
Min-Access: read-only</td><td>none</td></tr>
<tr><td>ifPromiscuous Mode</td><td>none</td><td>Specifies that write access is not required.
Min-Access: read-only</td><td>none</td></tr>
<tr><td>ifStackStatus</td><td>Integer {active(1)} (a subset of RowStatus)</td><td>Specifies that write access is not required. One of the six enumerated values for the RowStatus textual convention must be supported, specifically active(1).
Min-Access: read-only</td><td>none</td></tr>
</table>

Table 2-64 Interfaces Group Conformance MIB Objects (continued)

Name	Syntax	Description	Default Value
ifAdminStatus	`Integer {up(1), down(2)}`	Write access is not required or supported for the value `testing(3)`. Min-Access: read-only	none
ifAlias	none	Specifies that write access is not required. Min-Access: read-only	none
ifGeneral InformationGroup	`{ifIndex, ifDescr, ifType, ifSpeed, ifPhysAddress, ifAdminStatus, ifOperStatus, ifLastChange, ifLinkUpDownTrapEnable, ifConnectorPresent, ifHighSpeed, ifName, ifNumber, ifAlias, ifTableLastChange}`	Provides information applicable to all network interfaces for a a collection of objects. Status: current	none
ifFixedLength Group	`{ifInOctets, ifOutOctets, ifInUnknownProtos, ifInErrors, ifOutErrors}`	Provides information specific to either nonhigh speed (nonhigh speed interfaces transmit and receive at speeds less than or equal to 20 million bits/second), character-oriented, or fixed-length-transmission network interfaces. Status: current	none
ifHCFixedLength Group	`{ifHCInOctets, ifHCOutOctets, ifInOctets, ifOutOctets, ifInUnknownProtos, ifInErrors, ifOutErrors}`	Provides information specific to either high speed (greater than 20 million bits/second), character-oriented, or fixed-length-transmission network interfaces. Status: current	none
ifPacketGroup	`{ifInOctets, ifOutOctets, ifInUnknownProtos, ifInErrors, ifOutErrors, ifMtu, ifInUcastPkts, ifInMulticastPkts, ifInBroadcastPkts, ifInDiscards, ifOutUcastPkts, ifOutMulticastPkts, ifOutBroadcastPkts, ifOutDiscards, ifPromiscuousMode}`	Provides information specific to nonhigh speed (nonhigh speed interfaces transmit and receive at speeds less than or equal to 20 million bits/second), packet-oriented network interfaces. Status: current	none

Table 2-64 Interfaces Group Conformance MIB Objects (continued)

Name	Syntax	Description	Default Value
ifHCPacketGroup	{ifHCInOctets, ifHCOutOctets, ifInOctets, ifOutOctets, ifInUnknownProtos, ifInErrors, ifOutErrors, ifMtu, ifInUcastPkts, ifInMulticastPkts, ifInBroadcastPkts, ifInDiscards, ifOutUcastPkts, ifOutMulticastPkts, ifOutBroadcastPkts, ifOutDiscards, ifPromiscuousMode}	Provides information specific to high-speed (greater than 20 million bits/second but less than or equal to 650 million bits/second), packet-oriented network interfaces. Status: current	none
ifVHCPacketGroup	{ifHCInUcastPkts, ifHCInMulticastPkts, ifHCInBroadcastPkts, ifHCOutUcastPkts, ifHCOutMulticastPkts, ifHCOutBroadcastPkts, ifHCInOctets, ifHCOutOctets, ifInOctets, ifOutOctets, ifInUnknownProtos, ifInErrors, ifOutErrors, ifMtu, ifInUcastPkts, ifInMulticastPkts, ifInBroadcastPkts, ifInDiscards, ifOutUcastPkts, ifOutMulticastPkts, ifOutBroadcastPkts, ifOutDiscards, ifPromiscuousMode}	Provides information specific to higher-speed (greater than 650 million bits/second), packet-oriented network interfaces. Status: current	none
ifRcvAddressGroup	{ifRcvAddressStatus, ifRcvAddressType}	Provides information on multiple addresses that an interface receives. Status: current	none
ifStackGroup2	{ifStackStatus, ifStackLastChange}	Provides information on the layering of MIB-II interfaces. Status: current	none
ifCounter DiscontinuityGroup	{ifCounterDiscontinuity Time}	Provides information specific to interface discontinued counter values. Status: current	none

RS-232 MIB Objects

The RS-232 MIB module is used for hardware devices and resides in the RS-232-MIB.my file.

Note RS-232 objects are used only for PXM45 and PXM1.

The RS-232 MIB objects include:

- RS-232 Number
- RS-232 General Port Table
- RS-232 Asynchronous Port Table
- RS-232 Conformance and Compliance Statements

RS-232 Number

The object identifier for the MIB object is listed in Table 2-65.

Table 2-65 RS-232 Object Identifier

Name	Object Identifier
rs232Number	::= { rs232 1}

The MIB object is listed in Table 2-66.

Table 2-66 RS-232 MIB Object

Name	Syntax	Description	Default Value
rs232Number	`Integer32`	Determines the number of ports, regardless of their current state, used in the RS-232-like general port table. Max-Access: read-only Status: current	none

RS-232 General Port Table

The object identifier for each MIB object is listed in Table 2-67.

Table 2-67 RS-232 General Port Table Object Identifiers

Name	Object Identifier
rs232PortTable	::= { rs232 2 }
rs232PortEntry	::= { rs232PortTable 1 }
rs232PortIndex	::= { rs232PortEntry 1 }
rs232PortType	::= { rs232PortEntry 2 }

Table 2-67 RS-232 General Port Table Object Identifiers (continued)

Name	Object Identifier
rs232PortInSigNumber	::= { rs232PortEntry 3 }
rs232PortOutSigNumber	::= { rs232PortEntry 4 }
rs232PortInSpeed	::= { rs232PortEntry 5 }
rs232PortOutSpeed	::= { rs232PortEntry 6 }
rs232PortInFlowType	::= { rs232PortEntry 7 }
rs232PortOutFlowType	::= { rs232PortEntry 8 }

The MIB objects are listed in Table 2-68.

Table 2-68 RS-232 General Port Table MIB Objects

Name	Syntax	Description	Default Value
rs232PortTable	`SEQUENCE OF Rs232PortEntry`	Determines a list of port entries. The number of entries contains the value of `rs232Number`. Max-Access: not-accessible Status: current	none
rs232PortEntry	`Rs232PortEntry`	Determines the status and parameter values for a port. Max-Access: not-accessible Status: current The index is `rs232PortIndex`.	none
rs232PortIndex	`InterfaceIndex`	Determines the value of `ifIndex` for the port. By convention, hardware port numbers map directly to external connectors. The value for each port must remain constant at least from one reinitialization of the network management agent to the next. Max-Access: read-only Status: current	none
rs232PortType	`INTEGER { other(1), rs232(2), rs422(3), rs423(4), v35(5), x21(6) }`	Determines the hardware type for the port. Max-Access: read-only Status: current	none
rs232PortInSig Number	`Integer32`	Determines the number of input signals for the port in the `rs232PortInSigTable` input signal table. The `rs232PortInSigTable` contains entries only for those signals that the software detects are useful to observe. Max-Access: read-only Status: current	none

Table 2-68 RS-232 General Port Table MIB Objects (continued)

Name	Syntax	Description	Default Value
rs232PortOutSig Number	`Integer32`	Determines the number of output signals for the port in the `rs232PortOutSigTable` output signal table. `rs232PortOutSigTable` table contains entries only for those signals the software detects are useful to observe. Max-Access: read-only Status: current	none
rs232PortInSpeed	`Integer32`	Determines the input speed for the port in bps. **Note** Nonstandard values, for example, `9612`, probably are not allowed on most implementations. Max-Access: read-write Status: current	none
rs232PortOut Speed	`Integer32`	Determines output speed for the port in bps. **Note** Nonstandard values, for example, `9612`, probably are not allowed on most implementations. Max-Access: read-write Status: current	none
rs232PortInFlow Type	`INTEGER { none(1), ctsRts(2), dsrDtr(3) }`	Determines the input flow control for the port type. The following are the parameters: • `none`—No flow control at this level. • `ctsRts`—Usage of the hardware signals. • `dsrDtr`—Usage of the hardware signals. Max-Access: read-write Status: current	none
rs232PortOutFlow Type	`INTEGER { none(1), ctsRts(2), dsrDtr(3) }`	Determines the output flow control for the port type. The following are the parameters: • `none`—No flow control at this level. • `ctsRts`—Usage of the hardware signals. • `dsrDtr`—Usage of the hardware signals. Max-Access: read-write Status: current	none

RS-232 Asynchronous Port Table

The object identifier for each MIB object is listed in Table 2-69.

Table 2-69 RS-232 Asynchronous Port Table Object Identifiers

Name	Object Identifier
rs232AsyncPortTable	::= { rs232 3 }
rs232AsyncPortEntry	::= { rs232AsyncPortTable 1 }
rs232AsyncPortIndex	::= { rs232AsyncPortEntry 1 }
rs232AsyncPortBits	::= { rs232AsyncPortEntry 2 }
rs232AsyncPortStopBits	::= { rs232AsyncPortEntry 3 }
rs232AsyncPortParity	::= { rs232AsyncPortEntry 4 }
rs232AsyncPortAutobaud	::= { rs232AsyncPortEntry 5 }
rs232AsyncPortParityErrs	::= { rs232AsyncPortEntry 6 }
rs232AsyncPortFramingErrs	::= { rs232AsyncPortEntry 7 }
rs232AsyncPortOverrunErrs	::= { rs232AsyncPortEntry 8 }

The MIB objects are listed in Table 2-70.

Table 2-70 RS-232 Asynchronous Port Table MIB Objects

Name	Syntax	Description	Default Value
rs232AsyncPort Table	`SEQUENCE OF Rs232AsyncPortEntry`	Determines a list of asynchronous port entries. Entries do not need to exist for synchronous ports. Max-Access: not-accessible Status: current	none
rs232AsyncPort Entry	`Rs232AsyncPortEntry`	Determines both the status and parameter values for an asynchronous port. Max-Access: not-accessible Status: current The index is `rs232AsyncPortIndex`.	none
rs232AsyncPort Index	`InterfaceIndex`	Determines a unique value for each port. The value is the same as `rs232PortIndex` for the port. Max-Access: read-only Status: current	none
rs232AsyncPort Bits	`INTEGER (5..8)`	Determines the bits in a character for the port number. Max-Access: read-write Status: current	none

Table 2-70 RS-232 Asynchronous Port Table MIB Objects (continued)

Name	Syntax	Description	Default Value
rs232AsyncPort StopBits	`INTEGER { one(1), two(2), oneAndHalf(3), dynamic(4) }`	Determines the stop bits for the port number. Max-Access: read-write Status: current	none
rs232AsyncPort Parity	`INTEGER { none(1), odd(2), even(3), mark(4), space(5) }`	Determines the character parity bit for the port sense. Max-Access: read-write Status: current	none
rs232AsyncPort Autobaud	`INTEGER { enabled(1), disabled(2) }`	Determines the control to automatically sense input speed for the port ability. When `rs232PortAutoBaud` is set to `enabled`, a port can autobaud to values different from the set values for speed, parity, and character size. As a result, an NMS can temporarily observe values different from those previously set. Max-Access: read-write Status: current	none
rs232AsyncPort ParityErrs	`Counter32`	Determines the total number of characters with a parity error input from the port since system reinitialization. The port state is either set to `up` or `test`. Max-Access: read-only Status: current	none
rs232AsyncPort FramingErrs	`Counter32`	Determines the total number of characters with a framing error input from the port since system reinitialization. The port state is either set to `up` or `test`. Max-Access: read-only Status: current	none
rs232AsyncPort OverrunErrs	`Counter32`	Determines the total number of characters with an overrun error input from the port since system reinitialization. The port state is either set to `up` or `test`. Max-Access: read-only Status: current	none

RS-232 Conformance and Compliance Statements

RS-232 specific conformance information is listed in Table 2-71.

Table 2-71 RS-232 Conformance Groups

Name	Object Identifier
rs232Conformance	::= { rs232 7 }
rs232Groups	::= { rs232Conformance 1 }
rs232Compliances	::= { rs232Conformance 2 }

RS-232 Compliance Statements

The groups used for compliance statements are listed in Table 2-72.

Table 2-72 RS-232 MIB Objects Used for Compliance Statements

Name	Syntax	Description	Default Value
rs232Compliance (module compliance)	none	Specifies that the compliance statement for SNMPv2 entities contain RS-232 hardware interfaces. **Note** `rs232Group` is the mandatory group for the `rs232Compliance` module. Status: current	none
rs232AsyncGroup	none	Specifies that the asynchronous group is mandatory for SNMPv2 entities that have asynchronous RS-232 interfaces.	none

RS-232 Units of Conformance

The object identifiers for each MIB object are listed in Table 2-73.

Table 2-73 RS-232 Units of Conformance Object Identifiers

Name	Object Identifier
rs232Group	::= { rs232Groups 1 }
rs232AsyncGroup	::= { rs232Groups 2 }

The objects used for units of conformance are listed in Table 2-74.

Table 2-74 RS-232 MIB Objects Used for Units of Conformance

Name	Objects	Description	Default Value
rs232Group	rs232Number, rs232PortIndex, rs232PortType, rs232PortInSigNumber, rs232PortOutSigNumber, rs232PortInSpeed, rs232PortOutSpeed, rs232PortInFlowType, rs232PortOutFlowType, rs232InSigPortIndex, rs232InSigName, rs232InSigState, rs232InSigChanges, rs232OutSigPortIndex, rs232OutSigName, rs232OutSigState, rs232OutSigChanges	Determines a collection of objects that provides information applicable to all RS-232 interfaces. Status: current	none
rs232AsyncGroup	rs232AsyncPortIndex, rs232AsyncPortBits, rs232AsyncPortStopBits, rs232AsyncPortParity, rs232AsyncPortAutobaud, rs232AsyncPortParityErrs, rs232AsyncPortFramingErrs, rs232AsyncPortOverrunErrs	Determines a collection of objects that provides information applicable to asynchronous RS-232 interfaces. Status: current	none

CHAPTER 3

Cisco Enterprise MIB Objects

This chapter describes the common MIB objects used for Cisco enterprise that are used for AXSM, AXSM-E, PXM45, and RPM.

Contents of this chapter include:

- ATM Virtual Interface
- ATM Cell Layer
- Cisco DS3
- Cisco SONET
- Cisco WAN SONET
- Cisco WAN Resource Partition
- Cisco WAN ATM Connection
- Cisco WAN Feeder
- Cisco WAN Module
- Cisco Bulk File
- Cisco WAN Statistics Collection Manager
- Cisco System
- ATM Connection Statistics
- Robust Trap Mechanism MIB Objects
- Error Status MIB Objects
- Basis Shelf MIB Objects
- Generic MIB for Traps
- Module MIB for Traps
- Service Class Template
- Cisco WAN Class of Service Buffer

ATM Virtual Interface

This section describes the MIB objects used to create the ATM virtual Interface. The ATM Virtual Interface Objects are defined in the file CISCO-ATM-VIRTUAL-IF MIB.my.

Note The ATM Virtual Interface (IF) MIB is supported by AXSM and AXSM-E.

The MIB objects used for the ATM virtual interface include:

- ATM Virtual Interface Configuration Group
- ATM Virtual Interface Egress Statistics Table
- ATM Virtual Interface Egress Interval Statistics Table
- ATM Virtual Interface Ingress Statistics Table
- ATM Virtual Interface Conformance Information

ATM Virtual Interface Configuration Group

The object identifier for each MIB object is listed in Table 3-1.

Table 3-1 ATM Virtual Interface Configuration Group Object Identifiers

Name	Object Identifier
caviTable	::= {caviConfig 1}
caviEntry	::= {caviTable 1}
caviIndex	::= {caviEntry 1}
caviPhyIfIndex	::= {caviEntry 2}
caviViIfIndex	::= {caviEntry 3}
caviMinRate	::= {caviEntry 4}
caviMaxRate	::= {caviEntry 5}
caviFileId	::= {caviEntry 6}
caviIfType	::= {caviEntry 7}
caviVpiNum	::= {caviEntry 8}
caviRowStatus	::= {caviEntry 9}

A list of the MIB objects is shown in Table 3-2.

Table 3-2 ATM Virtual Interface Configuration Group MIB Objects

Name	Syntax	Description	Default Value
caviTable	Sequence of caviEntry	Creates physical interfaces that are used for the ATM virtual interfaces. For each entry created in this table, an entry will be created in the ifTable with ifType atmVirtual(149). For each entry deleted from this table, the corresponding entry from ifTable will be deleted. The ifIndex value for ifType atmVirtual(149) is used for indexing to atmInterfaceConfTable for getting information on an ATM virtual interface. Max-Access: not-accessible Status: current Dependencies/Restrictions: The parameters cannot be changed if there are one or more resource partitions configured on the interface.	none
caviEntry	caviEntry	Provides an entry in the caviTable table for each ATM virtual interface. This entry is created only if values for all objects in a row are provided. The caviVpiNum value is required only if caviIfType is vuni(4) or vnni(5). Max-Access: not-accessible Status: current Created By: Manager The index contains caviIndex. The referential integrity constraints are as follows: • The value of caviPhyIfIndex is the same as the ifIndex value in the ifTable. This instance must have its ifAdminStatus value as Up(1) before an entry can be created in the caviTable. • The SCT file specified by caviFileId must exist on the PXM disk before an entry can be created. • An entry for caviVpiNum is required when caviIfType has the value vnni(3). • Multiple virtual interfaces can be added to the same line if the value of caviIfType is vnni. Each vnni virtual interface is then identified by its unique vpi value, which must be specified in caviVpiNum. If the value of caviIfType is uni or nni, only a single virtual interface can be added to a line. • The value of caviRowStatus can be set to destroy (6) only if there are no entries in the cwRsrcPartConfTable for the corresponding ifIndex value.	none

Table 3-2 ATM Virtual Interface Configuration Group MIB Objects (continued)

<table>
<tr><th>Name</th><th>Syntax</th><th>Description</th><th>Default Value</th></tr>
<tr><td>caviIndex</td><td>Integer32 (1..2147483647)</td><td>Provides a unique value for the ATM virtual interface.
Max-Access: not accessible
Status: current
The following fields are used for the index entry format:
<table>
<tr><th>Field</th><th>Description</th><th>Bit Positions</th></tr>
<tr><td>Chassis #</td><td>Shelf Number of the module that is plugged in.</td><td>24 to 31 (eight bits)</td></tr>
<tr><td>Logical Slot #</td><td>Logical Slot Number of the module that is plugged in.</td><td>16 to 23 (eight bits)</td></tr>
<tr><td>Value</td><td>(See Table 3-1.)</td><td>0 to 15 (refers to a number between 1–60)</td></tr>
</table></td><td>none</td></tr>
<tr><td>caviPhyIfIndex</td><td>InterfaceIndex</td><td>Identifies the interface number ifIndex assigned to SONET(39), ds3(30), ds1(18), atmIma(107).
Max-Access: read-create
Status: current</td><td>none</td></tr>
<tr><td>caviViIfIndex</td><td>InterfaceIndex</td><td>Identifies the ATM virtual interface number ifIndex assigned to this entry, which is used to identify corresponding rows in the IF-MIB.
Note Reinitialization of the management agent can cause a client’s caviViIfIndex to change.
For information on the Chassis/Slot (CS) format used for this index entry, see caviIndex.
Max-Access: read-only
Status: current</td><td>none</td></tr>
</table>

Table 3-2 ATM Virtual Interface Configuration Group MIB Objects (continued)

<table>
<tr><th>Name</th><th>Syntax</th><th>Description</th><th>Default Value</th></tr>
<tr><td>caviMinRate</td><td>Unsigned32</td><td>Describes the ATM virtual interface's minimum cell rate (MCR). This is the guaranteed bandwidth allocated for the interface. The sum of caviMinRate of all ATM virtual interfaces configured on a single physical line must not exceed the bandwidth of the physical line. Creation of a row is rejected when the sum of caviMinRate of all existing ATM virtual interfaces exceeds the physical line bandwidth. The value of this object cannot exceed the ifSpeed or ifHighSpeed value in ifTable for the associated caviPhyIfIndex.
Units: cells-per-second
Max-access: read-create
Status: current
The various physical lines used for the ATM cell bandwidths are as follows:
<table>
<tr><th>Line Type</th><th>Bandwidth (cells per second)</th></tr>
<tr><td>OC192</td><td>22605280</td></tr>
<tr><td>OC48</td><td>5651328</td></tr>
<tr><td>OC12</td><td>1412832</td></tr>
<tr><td>OC3</td><td>353208</td></tr>
<tr><td>T3</td><td>96000</td></tr>
<tr><td>E3</td><td>80000</td></tr>
</table></td><td>none</td></tr>
<tr><td>caviMaxRate</td><td>Unsigned32</td><td>Describes the ATM virtual interface maximum cell rate. If bandwidth is available, the interface will be allocated bandwidth up to this value. This is the maximum bandwidth allocated for the interface. The total bandwidth of connections configured over this interface cannot exceed this value. The value of this object cannot exceed the ifSpeed or ifHighSpeed value in ifTable for the associated caviPhyIfIndex.
For information about the ATM cell bandwidths for various physical lines, see caviMinRate.
Units: cells-per-second
Max-Access: read-create
Status: current</td><td>none</td></tr>
<tr><td>caviFileId</td><td>Unsigned32</td><td>Determines the ID of the file that holds module specific configuration parameters for this ATM Virtual interface.
Max-Access: read-create
Status: current</td><td>0</td></tr>
</table>

Table 3-2 ATM Virtual Interface Configuration Group MIB Objects (continued)

Name	Syntax	Description	Default Value
caviIfType	INTEGER {uni(1), nni(2), vnni(3), vuni(4)}	Reflects the format of the cells transmitted by the physical interface. The following formats are configured: • uni:user port • nni:network port • vnni:virtual network port • vuni:virtual user port **Note** The vuni format is not supported. Max-Access: read-create Status: current	uni
caviVpiNum	AtmVpIdentifier	Configures this ATM virtual interface to be a virtual trunk. A nonzero value is required for virtual trunk operation. A value of 0 means that the ATM virtual interface is configured as a user port or a network port. Max-Access: read-create Status: current	0
caviRowStatus	RowStatus	Allows create and delete operations on caviTable entries. An entry is created in the table by setting this object to createAndGo(4). An entry is deleted from the table by setting this object to destroy(6). An entry in this table is modified after creation. Max-Access: read-create Status: current	none

ATM Virtual Interface Egress Statistics Table

The object identifiers used for the ATM Virtual Interface Egress Statistics Table are listed in Table 3-3.

Table 3-3 ATM Virtual Interface Egress Statistics Table Object Identifiers

Name	Object Identifier
caviStatEgressTable	::= {caviStatistics 1}
caviStatEgressEntry	::= {caviStatTable 1}
caviEgrRcvClp0Cells	::= {caviStatEgressEntry 1}
caviEgrRcvClp1Cells	::= {caviStatEgressEntry 2}
caviEgrClp0DiscCells	::= {caviStatEgressEntry 3}
caviEgrClp1DiscCellsE	::= {caviStatEgressEntry 4}
caviEgrXmtClp0Cells	::= {caviStatEgressEntry 5}
caviEgrXmtClp1Cells	::= {caviStatEgressEntry 6}
caviEgrRcvOAMCells	::= {caviStatEgressEntry 7}
caviEgrRMCells	::= {caviStatEgressEntry 8}

Table 3-3 ATM Virtual Interface Egress Statistics Table Object Identifiers (continued)

Name	Object Identifier
caviEgrRcvEFCICells	::= {caviStatEgressEntry 9}
caviEgrRcvEFCICells	::= {caviStatEgressEntry 10}
caviEgrXmtOAMCells	::= {caviStatEgressEntry 11}
caviHEgrXmtClp0Cells	::= {caviStatEgressEntry 12}
caviHEgrXmtClp1Cells	::= {caviStatEgressEntry 13}

The MIB objects are listed in Table 3-4.

Table 3-4 ATM Virtual Interface Egress Statistics Table MIB Objects

Name	Syntax	Description	Default Value
caviStatEgressTable	`Sequence of caviStatEntry`	Reflects real-time statistics associated with each of the ATM virtual interfaces on the egress side. Ingress direction means coming from the network into the switch; egress direction means going from the switch into the network. Max-Access: not-accessible Status: current	none
caviStatEgressEntry	`caviStatEgressEntry`	Provides an entry for egress ATM virtual interface statistics. The following are the descriptions: • `CLP-0` means Cell Loss Priority = 0 • `CLP-1` means Cell Loss Priority = 1 Statistics are collected before and after the traffic management and policing device, in both ingress and egress directions. Max-Access: not-accessible Status: current The index is `ifIndex`.	none
caviEgrRcvClp0Cells	`Counter32`	Specifies the number of `CLP-0` cells received from the switch at the traffic management and policing device. Max-Access: read-only Status: current	none
caviEgrRcvClp1Cells	`Counter32`	Specifies the number of `CLP-1` cells received from the switch at the traffic management and policing device. Max-Access: read-only Status: current	none
caviEgrClp0DiscCells	`Counter32`	Specifies the number of `CLP-0` cells discarded due to this policing device. Max-Access: read-only Status: current	none

Table 3-4 ATM Virtual Interface Egress Statistics Table MIB Objects (continued)

Name	Syntax	Description	Default Value
caviEgrClp1DiscCellsE	Counter32	Specifies the number of CLP-1 cells discarded as a result of policing. Max-Access: read-only Status: current	none
caviEgrXmtClp0Cells	Counter32	Specifies the number of CLP-0 cells transmitted to the network after traffic management and policing. Max-Access: read-only Status: current	none
caviEgrXmtClp1Cells	Counter32	Specifies the number of CLP-1 cells transmitted to the network after traffic management and policing. Max-Access: read-only Status: current	none
caviEgrRcvOAMCells	Counter32	Specifies the number of Operation and Maintenance (OAM) cells received from the switch at the traffic management and policing device. Max-Access: read-only Status: current	none
caviEgrRMCells	Counter32	Determines the number of Resource Management (RM) cells transmitted to the network from the switch. Max-Access: read-only Status: current	none
caviEgrRcvEFCICells	Counter32	Determines the number of Explicit Forward Congestion Indication (EFCI) cells transmitted to the network after traffic management and policing. Max-Access: read-only Status: current	none
caviEgrRcvEFCICells	Counter32	Determines the number of EFCI cells received from the switch at the traffic management and policing device. Max-Access: read-only Status: current	none
caviEgrXmtOAMCells	Counter32	Determines the number of OAM loopback cells transmitted to the network after traffic management and policing. Max-Access: read-only Status: current	none

Table 3-4 ATM Virtual Interface Egress Statistics Table MIB Objects (continued)

Name	Syntax	Description	Default Value
caviHEgrXmtClp0Cells	`Counter32`	Specifies the 64-bit version of `caviEgrXmtClp0Cells`. Max-Access: read-only Status: current	none
caviHEgrXmtClp1Cells	`Counter32`	Specifies the 64-bit version of `caviEgrXmtClp1Cells`. Max-Access: read-only Status: current	none

ATM Virtual Interface Egress Interval Statistics Table

The object identifiers are listed in Table 3-5.

Table 3-5 ATM Virtual Interface Egress Interval Statistics Table Object Identifiers

Name	Object Identifier
caviEgressIntervalTable	::= {caviStatistics 2}
caviEgressIntervalEntry	::= {caviEgressIntervalTable 1}
caviEgressIntervalNumber	::= {caviEgressIntervalEntry 1}
caviIntEgrRcvClp0Cells	::= {caviEgressIntervalEntry 2}
caviIntEgrRcvClp1Cells	::= {caviEgressIntervalEntry 3}
caviIntEgrClp0DiscCells	::= {caviEgressIntervalEntry 4}
caviIntEgrClp1DiscCells	::= {caviEgressIntervalEntry 5}
caviIntEgrXmtClp0Cells	::= {caviEgressIntervalEntry 6}
caviIntEgrXmtClp1Cells	::= {caviEgressIntervalEntry 7}
caviIntEgrRcvOAMCells	::= {caviEgressIntervalEntry 8}
caviIntEgrRMCells	::= {caviEgressIntervalEntry 9}
caviIntEgrXmtEFCICells	::= {caviEgressIntervalEntry 10}
caviIntEgrRcvEFCICells	::= {caviEgressIntervalEntry 11}
caviIntEgrXmtOAMCells	::= {caviEgressIntervalEntry 12}

The MIB objects are listed in Table 3-6.

Table 3-6 ATM Virtual Interface Egress Interval Statistics Table MIB Objects

Name	Syntax	Description	Default Value
caviEgressIntervalTable	`Sequence of caviEgressIntervalEntry`	Reflects the interval statistics associated with each of the ATM virtual interfaces in the egress direction. Max-Access: not-accessible Status: current	none
caviEgressInterval Entry	`caviEgressIntervalEntry`	Provides an entry for egress ATM Virtual Interface interval statistics. In addition to the current 15-minute interval bucket, the previous 24 hours of 15-minute interval buckets are collected for each ATM virtual interface. Before and after traffic management and policing device, statistics are being collected in egress direction. Max-Access: not-accessible Status: current The following are the indexes: • `ifIndex` • `caviEgressIntervalNumber`	none
caviEgressInterval Number	`Unsigned32(0..96)`	Contains a number used to uniquely identify the ATM virtual interface interval statistics. The current 15-minute interval is identified by 0. The previous 24 hours of 15-minute interval buckets are identified by 1 to 96. Max-Access: read-only Status: current	none
caviIntEgrRcvClp0Cells	`Counter32`	Specifies the number of `CLP-0` cells received from the switch at the traffic management and policing device during a 15-minute interval. Max-Access: read-only Status: current	none
caviIntEgrRcvClp1Cells	`Counter32`	Specifies the number of `CLP-1` cells received from the switch at the traffic management and policing device during a 15-minute interval. Max-Access: read-only Status: current	none
caviIntEgrClp0Disc Cells	`Counter32`	Specifies the number of `CLP-0` cells discarded due to policing during a 15-minute interval. Max-Access: read-only Status: current	none
caviIntEgrClp1Disc Cells	`Counter32`	Specifies the number of `CLP-1` cells discarded due to policing during a 15-minute interval. Max-Access: read-only Status: current	none

Table 3-6 ATM Virtual Interface Egress Interval Statistics Table MIB Objects (continued)

Name	Syntax	Description	Default Value
caviIntEgrXmtClp0 Cells	Counter32	Specifies the number of CLP-0 cells transmitted to the network after traffic management and policing during a 15-minute interval. Max-Access: read-only Status: current	none
caviIntEgrXmtClp1 Cells	Counter32	Specifies the number of CLP-1 cells transmitted to the network after traffic management and policing during a 15-minute interval. Max-Access: read-only Status: current	none
caviIntEgrRcvOAM Cells	Counter32	Specifies the number of OAM cells received from the switch at the traffic management and policing device during a 15-minute interval. Max-Access: read-only Status: current	none
caviIntEgrRMCells	Counter32	Specifies the number of RM cells transmitted to the network from the switch during a 15-minute interval. Max-Access: read-only Status: current	none
caviIntEgrXmtEFCI Cells	Counter32	Specifies the number of EFCI cells transmitted to the network after traffic management and policing during a 15-minute interval. Max-Access: read-only Status: current	none
caviIntEgrRcvEFCI Cells	Counter32	Specifies the number of EFCI cells received from the switch at traffic management and policing device during a 15-minute interval. Max-Access: read-only Status: current	none
caviIntEgrXmtOAM Cells	Counter32	Specifies the number of OAM loopback cells transmitted to the network after traffic management and policing during a 15-minute interval. Max-Access: read-only Status: current	none

ATM Virtual Interface Ingress Statistics Table

The object identifiers are listed in Table 3-7.

Table 3-7 ATM Virtual Interface Ingress Interval Statistics Table Object Identifiers

Name	Object Identifier
caviStatIngressTable	::= {caviStatistics 3}
caviStatIngressEntry	::= {caviStatIngressTable 1}
caviIngRcvClp0Cells	::= {caviStatIngressEntry 1}
caviIngRcvClp1Cells	::= {caviStatIngressEntry 2}
caviIngClp0DiscCells	::= {caviStatIngressEntry 3}
caviIngClp1DiscCells	::= {caviStatIngressEntry 4}
caviIngXmtClp0Cells	::= {caviStatIngressEntry 5}
caviIngXmtClp1Cells	::= {caviStatIngressEntry 6}
caviIngRcvOAMCells	::= {caviStatIngressEntry 7}
caviIngRMCells	::= {caviStatIngressEntry 8}
caviIngXmtEFCICells	::= {caviStatIngressEntry 9}
caviIngRcvEFCICells	::= {caviStatIngressEntry 10}
caviIngXmtOAMCells	::= {caviStatIngressEntry 11}
caviHIngRcvClp0Cells	::= {caviStatIngressEntry 12}
caviHIngRcvClp1Cells	::= {caviStatIngressEntry 13}

The MIB objects are listed in Table 3-8.

Table 3-8 ATM Virtual Interface Ingress Interval Statistics Table MIB Objects

Name	Syntax	Description	Default Value
caviStatIngressTable	`Sequence of CaviStatIngressEntry`	Reflects real-time statistics associated with each of the ATM virtual interfaces on the ingress side. Max-Access: not-accessible Status: current	none
caviStatIngressEntry	`caviStatIngressEntry`	Provides an entry for egress ATM virtual interface statistics for the following descriptions: • `CLP-0` means Cell Loss Priority = 0 • `CLP-1` means Cell Loss Priority = 1 Before and after traffic management and policing device, statistics are being collected. Max-Access: not-accessible Status: current The index is `ifIndex`.	none

Table 3-8 ATM Virtual Interface Ingress Interval Statistics Table MIB Objects (continued)

Name	Syntax	Description	Default Value
caviIngRcvClp0Cells	Counter32	Specifies the number of CLP-0 cells received from the switch at the traffic management and policing device in ingress direction. Max-Access: read-only Status: current	none
caviIngRcvClp1Cells	Counter32	Specifies the number of CLP-1 cells received from network at the traffic management and policing device in ingress direction. Max-Access: read-only Status: current	none
caviIngClp0DiscCells	Counter32	Specifies the number of CLP-0 cells discarded due to policing in the ingress direction. Max-Access: read-only Status: current	none
caviIngClp1DiscCells	Counter32	Specifies the number of CLP-1 cells discarded due to policing in the ingress direction. Max-Access: read-only Status: current	none
caviIngXmtClp0Cells	Counter32	Specifies the number of CLP-0 cells transmitted to the switch after traffic management and policing in ingress direction. Max-Access: read-only Status: current	none
caviIngXmtClp1Cells	Counter32	Specifies the number of CLP-1 cells transmitted to the switch after traffic management and policing in ingress direction. Max-Access: read-only Status: current	none
caviIngRcvOAMCells	Counter32	Specifies the number of OAM cells received from the network at the traffic management and policing device in ingress direction. Max-Access: read-only Status: current	none
caviIngRMCells	Counter32	Specifies the number of RM cells transmitted to the switch from the network in ingress direction. Max-Access: read-only Status: current	none

Table 3-8 ATM Virtual Interface Ingress Interval Statistics Table MIB Objects (continued)

Name	Syntax	Description	Default Value
caviIngXmtEFCICells	Counter32	Specifies the number of EFCI cells transmitted to the switch after traffic management and policing in ingress direction. Max-Access: read-only Status: current	none
caviIngRcvEFCICells	Counter32	Specifies the number of EFCI cells received from the network at traffic management and policing device in the ingress direction. Max-Access: read-only Status: current	none
caviIngXmtOAMCells	Counter32	Specifies the number of OAM loopback cells transmitted to switch after traffic management and policing in ingress direction. Max-Access: read-only Status: current	none
caviHIngRcvClp0Cells	Counter64	Specifies the 64-bit version of caviIngRcvClp0Cells in the ingress direction. Max-Access: read-only Status: current	none
caviHIngRcvClp1Cells	Counter64	Specifies the 64-bit version of caviIngRcvClp1Cells in the ingress direction. Max-Access: read-only Status: current	none

ATM Virtual Interface Conformance Information

The information on conformance is SNMPv2-specific. The object identifiers are listed in Table 3-9.

Table 3-9 ATM Virtual Interface Conformance Groups

Name	Object Identifier
caviMIBCompliance	::= {caviMIBCompliances 1}
caviMIBGroup	::= {caviMIBGroups 1}
caviEgressStatMIBGroup and **caviStatMIBGroup**	::= {caviMIBGroups 2}
caviHighSpeedStatMIBGroup and **caviEgressHighSpeedStatMIBGroup**	::= {caviMIBGroups 3}
caviEgressIntervalMIBGroup	::={caviMIBGroups 4}
caviIngressStatMIBGroup	::= {caviMIBGroups 5}
caviIngressHighSpeedStatMIBGroup	::= {caviMIBGroups 6}

The objects are listed in Table 3-10.

Table 3-10 ATM Virtual Interface Objects Used for Conformance

<table>
<tr><th>Name</th><th>Objects</th><th>Description</th><th>Default Value</th></tr>
<tr><td>caviMIBCompliance</td><td>none</td><td>Specifies the compliance statement for Cisco AtmVirtualIf management group.
Module: AXSM
Status: current
The following are the mandatory groups for caviMIBCompliance:
• caviMIBGroup
• caviStatMIBGroup
The groups for caviMIBCompliance are as follows:
<table>
<tr><th>Group</th><th>Description</th></tr>
<tr><td>caviEgressHigh SpeedStatMIBGroup</td><td>This group is required if the system has interface speeds of OC12 or higher.</td></tr>
<tr><td>caviEgressInterval MIBGroup</td><td>This group is mandatory for the system that supports ATM Virtual Interface interval statistics.</td></tr>
<tr><td>caviIngressStatMIB Group</td><td>This group is mandatory for the system that supports ATM Virtual Interface Ingress statistics.</td></tr>
<tr><td>caviIngressHighSpeed StatMIBGroup</td><td>This group is mandatory for the system that supports ATM Virtual Interface Ingress statistics for interface speeds of OC12 or higher.</td></tr>
</table></td><td>none</td></tr>
<tr><td>caviMIBGroup</td><td>caviPhyIfIndex, caviViIfIndex, caviMinRate, caviMaxRate, caviFileId, caviIfType, caviVpiNum, caviRowStatus</td><td>Relates to ciscoAtmVirtualIf configuration group.
Status: current</td><td>none</td></tr>
<tr><td>caviEgressStatMIB Group</td><td>caviEgrRcvClp0Cells, caviEgrRcvClp1Cells, caviEgrClp0DiscCells, caviEgrClp1DiscCells, caviEgrXmtClp0Cells, caviEgrXmtClp1Cells, caviEgrRcvOAMCells, caviEgrRmCells, caviEgrXmtEFCICells, caviEgrRcvEFCICells, caviEgrXmtOAMCells</td><td>Relates to the ATM Virtual Interface statistics group caviStatistics.
Status: current</td><td>none</td></tr>
</table>

Table 3-10 ATM Virtual Interface Objects Used for Conformance (continued)

Name	Objects	Description	Default Value
caviHighSpeedStat MIBGroup	caviHIngRcvClp0Cells, caviHIngRcvClp1Cells, caviHEgrXmtClp0Cells, caviHEgrXmtClp1Cells	Applies to ATM Virtual Interfaces with the speed of OC12 or higher. Status: current	none
caviStatMIBGroup	caviIngRcvClp0Cells, caviIngRcvClp1Cells, caviIngClp0DiscCells, caviIngClp1DiscCells, caviIngXmtClp0Cells, caviIngXmtClp1Cells, caviIngXmtOAMCells, caviIngRMCells, caviIngRcvEFCICells, caviIngXmtEFCICells, caviIngRcvOAMCells, caviEgrRcvClp0Cells, caviEgrRcvClp1Cells, caviEgrClp0DiscCells, caviEgrClp1DiscCells, caviEgrXmtClp0Cells, caviEgrXmtClp1Cells, caviEgrRcvOAMCells, caviEgrRMCells, caviEgrXmtEFCICells, caviEgrRcvEFCICells, caviEgrXmtOAMCells	Relates to the ATM Virtual Interface statistics group caviStatistics. Status: current	none
caviEgressHigh SpeedStatMIBGroup	caviHEgrXmtClp0Cells, caviHEgrXmtClp1Cells	Applies to ingress ATM Virtual Interfaces with the speed of OC12 or higher. Status: current	none
caviEgressInterval MIBGroup	caviEgressInterval Number, caviIntEgrRcvClp0Cells, caviIntEgrRcvClp1Cells, caviIntEgrClp0DiscCells, caviIntEgrClp1DiscCells, caviIntEgrXmtClp0Cells, caviIntEgrXmtClp1Cells, caviIntEgrRcvOAMCells, caviIntEgrRMCells, caviIntEgrXmtEFCICells, caviIntEgrRcvEFCICells, caviIntEgrXmtOAMCells	Relates to the ATM Virtual Interface interval statistics group. Status: current	none

Table 3-10 ATM Virtual Interface Objects Used for Conformance (continued)

Name	Objects	Description	Default Value
caviIngressStatMIB Group	caviIngRcvClp0Cells, caviIngRcvClp1Cells, caviIngClp0DiscCells, caviIngClp1DiscCells, caviIngXmtClp0Cells, caviIngXmtClp1Cells, caviIngRcvOAMCells, caviIngRMCells, caviIngXmtEFCICells, caviIngRcvEFCICells, caviIngXmtOAMCells	Relates to the ingress ATM Virtual Interface statistics. Status: current	none
caviIngressHigh SpeedStatMIBGroup	caviHIngRcvClp0CElls, caviHIngRcvClp1Cells	Applies to ingress ATM Virtual Interfaces with speed of OC12 or higher. Status: current	none

ATM Cell Layer

This section describes the ATM cell layer objects and the statistics of a physical line. These MIB objects are defined in the CISCO-ATM-CELL-LAYER-MIB.my file.

Note The ATM Cell Layer MIB is supported by AXSM and AXSM-E.

The MIB objects used for the ATM cell layer include:

- ATM Cell Layer Configuration Group
- ATM Cell Layer Statistics Group
- ATM Cell Layer Conformance Information

ATM Cell Layer Configuration Group

The object identifiers are listed in Table 3-11.

Table 3-11 ATM Cell Layer Configuration Group Object Identifiers

Name	Object Identifier
caclConfigTable	::= {caclConfig 1}
caclConfigEntry	::= {caclConfigTable 1}
caclNullCellHeader	::= {caclConfigEntry 1}
caclNullCellPayload	::= {caclConfigEntry 2}
caclHecCosetEnable	::= {caclConfigEntry 3}
caclPayloadScramblingEnable	::= {caclConfigEntry 4}

The MIB objects are listed in Table 3-12.

Table 3-12 ATM Cell Layer Configuration Group MIB Objects

Name	Syntax	Description	Default Value
caclConfigTable	`Sequence of caclConfigEntry`	Describes the ATM cell layer configuration table. An entry in this table is automatically created for each entry in the `ifTable` which has an `ifType of ATM(37)`, and which is layered on top of a SONET, DS3, or DS1 interface on the ATM switch. Max-Access: not-accessible Status: current	none
caclConfigEntry	`caclConfigEntry`	Provides an entry in the `caclConfigTable`. The table is indexed by the `ifIndex` of the `ATM(37)` interface. Max-Access: not-accessible Status: current The index is `ifIndex`.	none
caclNullCellHeader	`OCTET STRING (SIZE(4))`	Specifies the first four bytes of the ATM header to be used for null cells. Generally, idle cells are inserted into a cellstream as a mechanism for rate adaptation between the ATM data link layer and the physical layer. The header fields Generic flow control, Payload type, and Cell Loss Priority are specified in the value of this object. It applies to both transmitted as well as received cells. Refer to ITU-T Recommendation I.432. Max-Access: read-write Status: current	none

Table 3-12 ATM Cell Layer Configuration Group MIB Objects (continued)

Name	Syntax	Description	Default Value
caclNullCellPayload	`Unsigned32 (0..255)`	Defines the Null Cell Payload. The idle cell information field can be specified in this object. It applies to both transmitted as well as received cells. The default value is `0x6A`. Refer to ITU-T Recommendation I.432. Max-Access: read-write Status: current	`6Ah`
caclHecCosetEnable	`TruthValue`	If enabled, the algorithm of Coset Polynomial Addition is applied to do header error check calculations. It applies to transmitted as well as received cells. Refer to ITU-T Recommendation I.432. Max-Access: read-write Status: current	`true`
caclPayloadScrambling Enable	`TruthValue`	Indicates whether Payload Scrambling is enabled. Refer to ITU-T Recommendation I.432. Max-Access: read-write Status: current	`true`

ATM Cell Layer Statistics Group

The object identifiers are listed in Table 3-13.

Table 3-13 ATM Cell Layer Statistics Group Object Identifiers

Name	Object Identifier
caclStatsTable	::= {caclStats 1}
caclStatsEntry	::= {caclStatsTable 1}
caclInRcvCLP0Cells	::= {caclStatsEntry 1}
caclInRcvCLP1Cells	::= {caclStatsEntry 2}
caclInValidOAMCells	::= {caclStatsEntry 3}
caclInErrOAMCells	::= {caclStatsEntry 4}
caclInGfcCells	::= {caclStatsEntry 5}
caclInVpiVciErrCells	::= {caclStatsEntry 6}
caclInLastUnknVpi	::= {caclStatsEntry 7}
caclInLastUnknVci	::= {caclStatsEntry 8}
caclInXmtCLP0Cells	::= {caclStatsEntry 9}
caclInXmtCLP1Cells	::= {caclStatsEntry 10}
caclInValidRMCells	::= {caclStatsEntry 11}

Table 3-13 ATM Cell Layer Statistics Group Object Identifiers (continued)

Name	Object Identifier
caclInRcvIdleCells	::= {caclStatsEntry 12}
caclInHecErrDiscCells	::= {caclStatsEntry 13}
caclInHecErrCorrectedCells	::= {caclStatsEntry 14}
caclInUpcCLP0DiscCells	::= {caclStatsEntry 15}
caclInUpcTotalDiscCells	::= {caclStatsEntry 16}
caclInUpcTotalNonCompCells	::= {caclStatsEntry 17}
caclOutXmtCLP0Cells	::= {caclStatsEntry 18}
caclOutXmtCLP1Cells	::= {caclStatsEntry 19}
caclOutValidOAMCells	::= {caclStatsEntry 20}
caclOutErrOAMCells	::= {caclStatsEntry 21}
caclOutVpiVciErrCells	::= {caclStatsEntry 22}
caclOutRcvCLP0Cells	::= {caclStatsEntry 23}
caclOutRcvCLP1Cells	::= {caclStatsEntry 24}
caclOutRcvValidRMCells	::= {caclStatsEntry 25}
caclOutRcvIdleCells	::= {caclStatsEntry 26}
caclHInRcvCLP0Cells	::= {caclStatsEntry 27}
caclHInRcvCLP1Cells	::= {caclStatsEntry 28}
caclHOutXmtCLP0Cells	::= {caclStatsEntry 29}
caclHOutXmtCLP1Cells	::= {caclStatsEntry 30}

The MIB objects are listed in Table 3-14.

Table 3-14 ATM Cell Layer Statistics Group MIB Objects

Name	Syntax	Description	Default Value
caclStatsTable	`Sequence of CaclStatsEntry`	Describes the ATM Cell layer statistics table for the cell layer over a physical line. There is an entry for each SONET, DS3, or DS1 line. The table maintains objects that apply to ATM end-systems, as well as objects that apply only to ATM switches. The following is the descriptions of the objects: • `CLP-0` means Cell Loss Priority = 0 • `CLP-1` means Cell Loss Priority = 1 ATM cells with `CLP=0` have a higher priority in regard to cell loss than ATM cells with `CLP=1`. Therefore, during resource congestions, `CLP=1` cells are dropped before any `CLP=0` cell is dropped. Max-Access: not-accessible Status: current	none
caclStatsEntry	`caclStatsEntry`	Provides an entry for the ATM cell layer statistics for a physical line. These are real-time statistics, which are collected per cell layer over a physical line. Entries are created automatically for each line. The table is indexed by the `ifIndex` value of the `ATM(37)` interface. Max-Access: not-accessible Status: current The index is `ifIndex`.	none
caclInRcvCLP0Cells	`Counter32`	Determines the number of cells received on the interface with `CLP0` bit set. Max-Access: read-only Status: current	none
caclInRcvCLP1Cells	`Counter32`	Determines the number of cells received on the interface with `CLP1` bit set. Max-Access: read-only Status: current	none
caclInValidOAMCells	`Counter32`	Determines the number of valid OAM cells received on the interface. Max-Access: read-only Status: current	none
caclInErrOAMCells	`Counter32`	Determines the number of errored OAM cells received on the interface. Max-Access: read-only Status: current	none

Table 3-14 ATM Cell Layer Statistics Group MIB Objects (continued)

Name	Syntax	Description	Default Value
caclInGfcCells	`Counter32`	Determines the number of nonzero GFC cells received on the interface. Max-Access: read-only Status: current	none
caclInVpiVciErrCells	`Counter32`	Determines the number of cells received from the interface with unknown VPI/VCI values. Max-Access: read-only Status: current	none
caclInLastUnknVpi	`Integer32 (0..4095)`	Determines the last unknown VPI value. This object is valid only if `caclInVpiVciErrCells` is nonzero. Max-Access: read-only Status: current	none
caclInLastUnknVci	`Integer32 (0..65535)`	Determines the last unknown VCI value. This object is valid only if `caclInVpiVciErrCells` is nonzero. Max-Access: read-only Status: current	none
caclInXmtCLP0Cells	`Counter32`	Determines the number of `CLP0` cells received from the interface and transmitted to the switch fabric. Max-Access: read-only Status: current	none
caclInXmtCLP1Cells	`Counter32`	Determines the number of `CLP1` cells received from the interface and transmitted to the switch fabric. Max-Access: read-only Status: current	none
caclInValidRMCells	`Counter32`	Determines the number of valid RM cells received from the interface. Max-Access: read-only Status: current	none
caclInRcvIdleCells	`Counter32`	Determines the number of idle cells received from the interface. Max-Access: read-only Status: current	none
caclInHecErrDiscCells	`Counter32`	Provides error detection and correction from the ATM cell header by using the header error check (HEC) calculation. This object is the number of received cells that were discarded because they had HEC errors. Max-Access: read-only Status: current	none

Table 3-14 ATM Cell Layer Statistics Group MIB Objects (continued)

Name	Syntax	Description	Default Value
caclInHecErrCorrected Cells	Counter32	Specifies the number of received cells which had HEC errors that were corrected. Max-Access: read-only Status: current	none
caclInUpcCLP0DiscCells	Counter32	Allocates resources fairly among different users. Networks police traffic at resource access points. The traffic enforcement or policing taken at a UNI is called Usage Parameter Control (UPC). This is a counter associated with the number of CLP0 cells discarded due to policing. Max-Access: read-only Status: current	none
caclInUpcTotalDiscCells	Counter32	Total number of cells (CLP0+1) discarded due to policing. Max-Access: read-only Status: current	none
caclInUpcTotalNonComp Cells	Counter32	Determines the number of noncompliant cells (cells that are not compliant with a specific traffic profile) discarded due to policing. Max-Access: read-only Status: current	none
caclOutXmtCLP0Cells	Counter32	Determines the number of CLP0 cells transmitted to interface. Max-Access: read-only Status: current	none
caclOutXmtCLP1Cells	Counter32	Determines the number of CLP1 cells transmitted to interface. Max-Access: read-only Status: current	none
caclOutValidOAMCells	Counter32	Determines the number of OAM cells received by the interface from the switch fabric. Max-Access: read-only Status: current	none
caclOutErrOAMCells	Counter32	Determines the number of errored OAM cells received by the interface from the switch fabric. Max-Access: read-only Status: current	none

Table 3-14 ATM Cell Layer Statistics Group MIB Objects (continued)

Name	Syntax	Description	Default Value
caclOutVpiVciErrCells	Counter32	Determines the number of cells with unknown VPI or VCI value received by the interface from the switch fabric. Max-Access: read-only Status: current	none
caclOutRcvCLP0Cells	Counter32	Determines the number of CLP0 cells received by the interface from the switch fabric. Max-Access: read-only Status: current	none
caclOutRcvCLP1Cells	Counter32	Determines the number of CLP1 cells received by the interface from the switch fabric. Max-Access: read-only Status: current	none
caclOutRcvValidRM Cells	Counter32	Determines the number of valid RM cells received by the interface from the switch fabric. Max-Access: read-only Status: current	none
caclOutRcvIdleCells	Counter32	Determines the number of idle cells received by the interface from the switch fabric. Max-Access: read-only Status: current	none
caclHInRcvCLP0Cells	Counter64	Determines the 64-bit version of caclInRcvCLP0Cells. Max-Access: read-only Status: current	none
caclHInRcvCLP1Cells	Counter64	Determines the 64-bit version of caclnRcvCLP1Cells. Max-Access: read-only Status: current	none
caclHOutXmtCLP0Cells	Counter64	Determines the 64-bit version of caclOutXmtCLP0Cells. Max-Access: read-only Status: current	none
caclHOutXmtCLP1Cells	Counter64	Determines the 64-bit version of caclOutXmtCLP1Cells. Max-Access: read-only Status: current	none

ATM Cell Layer Conformance Information

One object identifier is listed in Table 3-15.

Table 3-15 ATM Cell Layer Conformance Information Object Identifier

Name	Object Identifier
caclMIBCompliance	::= {caclMIBCompliances 1}

One MIB object is listed in Table 3-16.

Table 3-16 ATM Cell Layer Conformance Information MIB Object

Name	Mandatory Groups	Description		Default Value
caclMIBCompliance	caclMIBGroup	Describes the compliance statement for the Cisco ATM Cell layer MIB. Status: current Module: AXSM The descriptions for each mandatory group include:		none
		Group	**Description**	
		caclATMSwitchStatsMIB Group	Required if the system is an ATM switch.	
		caclATMEndSyatemStatsMIB Group	Required if the system is an End System.	
		caclHighSpeedATMMIB Group	Required if the system has interface speed of OC12 or higher.	

ATM Cell Layer Units of Conformance

The object identifiers are listed in Table 3-17.

Table 3-17 ATM Cell Layer Units of Conformance Object Identifiers

Name	Object Identifier
caclMIBGroup	::= {caclMIBGroups 1}
caclATMSwitchStatsMIBGroup	::= {caclMIBGroups 2}
caclATMEndSyatemStatsMIBGroup	::= {caclMIBGroups 3}
caclHighSpeedATMMIBGroup	::= {caclMIBGroups 4}

The MIB objects are listed in Table 3-18.

Table 3-18 ATM Cell Layer Units of Conformance MIB Object

Name	Objects	Description	Default Value
caclMIBGroup	caclNullCellHeader, caclNullCellPayload, caclHecCosetEnable, caclPayloadScramblingEnable	Describes a collection of objects that provides ATM cell layer configuration parameters. Status: current	none
caclATMSwitchStats MIBGroup	caclInRcvCLP0Cells, caclInRcvCLP1Cells, caclInValidOAMCells, caclInErrOAMCells, caclInGfcCells, caclInVpiVciErrCells, caclInLastUnknVpi, caclInLastUnknVci, caclInXmtCLP0Cells, caclInXmtCLP1Cells, caclInValidRMCells, caclInRcvIdleCells, caclInHecErrDiscCells, caclInHecErrCorrectedCells, caclInUpcCLP0DiscCells, caclInUpcTotalDiscCells, caclInUpcTotalNonCompCells, caclOutXmtCLP0Cells, caclOutXmtCLP1Cells, caclOutValidOAMCells, caclOutErrOAMCells, caclOutVpiVciErrCells, caclOutRcvCLP0Cells, caclOutRcvCLP1Cells, caclOutRcvValidRMCells, caclOutRcvIdleCells	Describes a collection of objects that provides statistics for ATM switch interfaces. Status: current	none
caclATMEndSyatem StatsMIBGroup	caclInRcvCLP0Cells, caclInRcvCLP1Cells, caclInValidOAMCells, caclInErrOAMCells, caclInGfcCells, caclInVpiVciErrCells, caclInLastUnknVpi, caclInLastUnknVci, caclInValidRMCells, caclInRcvIdleCells, caclInHecErrDiscCells, caclInHecErrCorrectedCells, caclInUpcCLP0DiscCells, caclInUpcTotalDiscCells, caclInUpcTotalNonCompCells, caclOutXmtCLP0Cells, caclOutXmtCLP1Cells, caclOutValidOAMCells, caclOutErrOAMCells, caclOutVpiVciErrCells	Describes a collection of objects that provides statistics for ATM end-system interfaces. Status: current	none
caclHighSpeedATM MIBGroup	caclHInRcvCLP0Cells, caclHInRcvCLP1Cells, caclHOutXmtCLP0Cells, caclHOutXmtCLP1Cells	Applies to ATM interfaces with the speed of OC12 or higher. Status: current	none

Cisco DS3

This section describes the Cisco DS3 MIB objects defined in the CISCO-DS3-MIB.my file.

Note The Cisco DS3 MIB is supported by AXSM and AXSM-E.

The Cisco DS3 MIB groups include:

- DS3 Line Configuration Group
- DS3 Alarm Configuration Group
- DS3 Statistics Group
- DS3 PLCP Alarm Configuration Group
- DS3 PLCP Alarm Table
- DS3 PLCP Statistics Group
- Previous 24 Hour Interval Data
- Cisco DS3 Units of Conformance

DS3 Line Configuration Group

This group contains DS3 line configuration objects.

The object identifiers for each MIB object are listed in Table 3-19.

Table 3-19 DS3 Line Configuration Group Object Identifiers

Name	Object Identifier
cds3ConfigTable	::= {cds3Config 1}
cds3ConfigEntry	::= {cds3ConfigTable 1}
cds3LineType	::= {cds3ConfigEntry 1}
cds3LineAIScBitsCheck	::= {cds3ConfigEntry 2}
cds3LineRcvFEACValidation	::= {cds3ConfigEntry 3}
cds3LineOOFCriteria	::= {cds3ConfigEntry 4}
cds3TraceToTransmit	::= {cds3ConfigEntry 5}
cds3TraceToExpect	:= {cds3ConfigEntry 6}
cds3TraceAlarm	::= {cds3ConfigEntry 7}

A list of the MIB objects is shown in Table 3-20.

Table 3-20 DS3 Line Configuration Group MIB Objects

<table>
<tr><th>Name</th><th>Syntax</th><th>Description</th><th>Default Value</th></tr>
<tr><td>cds3Config Table</td><td>Sequence of cds3ConfigEntry</td><td>Contains objects to configure a T3/E3 line.
Max-Access: not-accessible
Status: current</td><td>none</td></tr>
<tr><td>cds3Config Entry</td><td>cds3ConfigEntry</td><td>Contains an entry in the table. There is an entry for each available T3/E3 line, with an ifType value of ds3(30). By default, lines are disabled. ifAdmin from ifTable must be used to up the line. The objects in this table are modified only when the line is up.
Max-Access: not-accessible
Status: current
Referential Integrity: The value of the ifAdminStatus object from ifTable for entries in this table can be set to down(2) only if there are no entries in caviTable for the corresponding caviPhyIfIndex value.
Created By: Agent
The index is ifIndex.</td><td>none</td></tr>
<tr><td>cds3LineType</td><td>INTEGER {ds3cbitadm(1), ds3cbitplcp(2), e3g832adm(3)}</td><td>Indicates the variety of DS3 C-bit or E3 application implementing this interface. The type of interface effects the interpretation of the usage and error statistics. The rate of DS3 is 44.736 Mbps and E3 is 34.368 Mbps.
Refer to ANSI T1.107, ANSI T1.102, CCITT G.751, and ETSI T/NA(91)18.
Max-Access: not-accessible
Status: current
The following values describe the three line type modes:
<table>
<tr><th>Value</th><th>Mode</th><th>Applicable Interfaces</th></tr>
<tr><td>ds3cbitadm</td><td>Indicates C-BIT ADM mode</td><td>Applies only to DS3 type interfaces</td></tr>
<tr><td>ds3cbitplcp</td><td>Indicates C-BIT PLCP mode</td><td>Applies only to DS3 type interfaces</td></tr>
<tr><td>e3g832adm</td><td>Indicates G.832 ADM mode</td><td>Applies only to E3 type interfaces</td></tr>
</table></td><td>none</td></tr>
<tr><td>cds3LineAISc BitsCheck</td><td>INTEGER {check(1), ignore(2)}</td><td>When check(1) is set, Alarm Indication Signal (AIS) is declared when 1010...b is found and C-Bits are all 0. When ignore(2) is set, an AIS condition is declared when the 1010...b is detected regardless of the state of the C-Bits.
Note This object does not apply to the E3 interface.
Refer to RFC 2496.
Max-Access: read-write
Status: current</td><td>none</td></tr>
</table>

Table 3-20 DS3 Line Configuration Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cds3LineRcvFEACValidation	INTEGER{fEACCodes4Of5(1), fEACCodes8Of10(2)}	Specifies the Far End Alarm and Control (FEAC) code validation criteria. If fEACCodes4Of5 is specified, a valid FEAC code is declared if 4 to 5 codes match. When fEACCodes8Of10 is specified, a valid FEAC code is declared when 8 of 10 codes match. **Note** This object does not apply to the E3 interface. Max-Access: read-write Status: current	fEACCodes4Of5
cds3LineOOFCriteria	INTEGER {bits3Of8(1), bits3Of16(2)}	Specifies the Out of Frame (OOF) decision criteria. If bits3Of8 is specified, an OOF condition is declared if at least 3 of 8 framing bits are in error. When bits3Of16 is specified, OOF is declared if 3 of 16 framing bits are found to be in error. When errors are detected in the framing octets for four consecutive frames, OOF is declared. **Note** This object does not apply to the E3 interface. Max-Access: read-write Status: current	bits3Of8
cds3TraceToTransmit	DisplayString (SIZE (1..16)	Transmits the G.832 Trail Trace string. Refer to Bellcore GR-253-CORE. Max-Access: read-write Status: current Dependencies/Restrictions: This object is not supported.	none
cds3TraceToExpect	DisplayString (SIZE (1..16)	Transmits the G.832 Trail Trace string that is expected to be received from the other end. Refer to Bellcore GR-253-CORE. Max-Access: read-write Status: current Dependencies/Restrictions: This object is not supported.	none
cds3TraceAlarm	INTEGER {noAlarm(1), traceFailure(2)}	Specifies the value of this object is set to traceFailure(2) when the DS3 line receives a trace that does not match the cds3TraceToExpect. Refer to Bellcore GR-253-CORE. Max-Access: read-only Status: current Dependencies/Restrictions: This object is not supported.	none

DS3 Alarm Configuration Group

The object identifiers are listed in Table 3-21.

Table 3-21 DS3 Alarm Configuration Group Object Identifiers

Name	Object Identifier
cds3AlarmConfigTable	::= {cds3AlarmConfig 1}
cds3AlarmConfigEntry	::= { cds3AlarmConfigTable 1 }
cds3NEAlarmUpCount	::= {cds3AlarmConfigEntry 1}
cds3NEAlarmDownCount	::= {cds3AlarmConfigEntry 2}
cds3NEAlarmThreshold	::= {cds3AlarmConfigEntry 3}
cds3FEAlarmUpCount	::= {cds3AlarmConfigEntry 4}
cds3FEAlarmDownCount	::= {cds3AlarmConfigEntry 5}
cds3FEAlarmThreshold	::= {cds3AlarmConfigEntry 6}
cds3StatisticalAlarmSeverity	::= {cds3AlarmConfigEntry 7}
cds3LCV15MinThreshold	::= {cds3AlarmConfigEntry 8}
cds3LCV24HrThreshold	::= {cds3AlarmConfigEntry 9}
cds3LES15MinThreshold	::= {cds3AlarmConfigEntry 10}
cds3LES24HrThreshold	::= {cds3AlarmConfigEntry 11}
cds3PCV15MinThreshold	::= {cds3AlarmConfigEntry 12}
cds3PCV24HrThreshold	::= {cds3AlarmConfigEntry 13}
cds3PES15MinThreshold	::= {cds3AlarmConfigEntry 14}
cds3PES24HrThreshold	::= {cds3AlarmConfigEntry 15}
cds3PSES15MinThreshold	::= {cds3AlarmConfigEntry 16}
cds3PSES24HrThreshold	::= {cds3AlarmConfigEntry 17}
cds3SEFS15MinThreshold	::= {cds3AlarmConfigEntry 18}
cds3SEFS24HrThreshold	::= {cds3AlarmConfigEntry 19}
cds3UAS15MinThreshold	::= {cds3AlarmConfigEntry 20}
cds3UAS24HrThreshold	::= {cds3AlarmConfigEntry 21}
cds3CCV15MinThreshold	::= {cds3AlarmConfigEntry 22}
cds3CCV24HrThreshold	::= {cds3AlarmConfigEntry 23}
cds3CES15MinThreshold	::= {cds3AlarmConfigEntry 24}
cds3CES24HrThreshold	::= {cds3AlarmConfigEntry 25}
cds3CSES15MinThreshold	::= {cds3AlarmConfigEntry 26}
cds3CSES24HrThreshold	::= {cds3AlarmConfigEntry 27}
cds3LineStatisticalAlarmStateE	::= {cds3AlarmConfigEntry 28}

The MIB objects are listed in Table 3-22.

Table 3-22 DS3 Alarm Configuration Group MIB Objects

Name	Syntax	Description	Default Value
cds3AlarmConfigTable	`Sequence of cds3AlarmConfigEntry`	Describes the DS3 interface alarm configuration table. This table contains the parameters associated with detecting and declaring alarms for the interface. The parameters include the severity of alarm, alarm integration parameters, and 15-minute and 24-hour thresholds. Max-Access: not-accessible Status: current	none
cds3AlarmConfigEntry	`Cds3AlarmConfigEntry`	Provides an entry in the T3/E3 Alarm Configuration Table. The entries are automatically created for an `ifType` value of `ds3(30)`, with default values for objects. Thresholds are configured for the following objects: • Line Coding Violations (LCV) • Line Errored Seconds (LES) • Line Severely Errored Seconds (LSES) • P-bit Coding Violations (PCV) • P-bit Errored Seconds (PES) • P-bit Severely Errored Seconds (PSES) • Severely Errored Seconds (SES) • Severely Errored Framing Seconds (SEFS) • Unavailable Seconds (UAS) • C-bit Coding Violations (CCV) • C-bit errored seconds (CES) Once these thresholds are exceeded, minor or major alarms are raised. Refer to RFC 2496 for descriptions of these errors. Max-Access: not-accessible Status: current The index is `ifIndex`.	none

Table 3-22 DS3 Alarm Configuration Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cds3NEAlarmUpCount	Unsigned32 (0..2147483647)	Specifies the increment to the Near End alarm integration counters. The local alarms are LOS and LOF. The default value is 6, and the integration counter is incremented by this value every 10 ms while the alarm persists. The counter is decremented by the value of cds3NEAlarmDownCount while there is no alarm (the counter does not decrement below 0). The integration counter exceeds an alarm of the severity, which is declared for the particular Near End alarm. Max-Access: read-write Status: current	6
cds3NEAlarmDown Count	Unsigned32 (0..2147483647)	Specifies the decrement to the Near End alarm integration counters. The alarms are LOS and LOF. The default value is 1, and the integration counter is decremented by this value every 10 milliseconds while no alarm persists. The counter is incremented by the value of cds3NEAlarmUpCount while there is no alarm (the counter does not decrement below 0). When the integration counter reaches 0, any alarm previously declared is removed. Max-Access: read-write Status: current	1
cds3NEAlarmThreshold	Unsigned32 (0..2147483647)	Specifies the value set in this object, which determines the value that the alarm integration counter must reach for an near end alarm to be declared. If the up count is 6, and the threshold 150, an alarm is integrated in 2.5 seconds. If the down count is 1, the alarm is cleared in 15 seconds. Max-Access: read-write Status: current	150
cds3FEAlarmUpCount	Unsigned32 (0..2147483647)	Specifies the increment to the Far End alarm integration counters. The alarms are Remote Alarm Indication (RAI) and Alarm Indication Signal (AIS). The integration counter is incremented by this value every 10 ms while the alarm persists. The counter is decremented by this value while there is no alarm (the counter does not decrement below 0). When the integration counter exceeds cds3FEAlarmThreshold, an alarm of the severity specified for the particular far end alarm is declared. Max-Access: read-write Status: current	6

Table 3-22 DS3 Alarm Configuration Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cds3FEAlarmDown Count	`Unsigned32 (0..2147483647)`	Specifies the decrement to the Far End alarm integration counters. The alarms are AIS and Remote Alarm Indication (RAI). While no alarm persists. the integration counter is decremented by this value every 10 milliseconds (till 0). The counter is incremented by this value while there is no alarm. When the integration counter decrements to 0, any previously declared alarm is removed. Max-Access: read-write Status: current	`1`
cds3FEAlarmThreshold	`Unsigned32 (0..2147483647)`	Specifies the value set in this object, which determines the value that the alarm integration counter must reach for and far end alarm to be declared. If the up count is 6, the down count is 1 and the threshold 150, an alarm is integrated in 2.5 seconds and removed in 15 seconds. Max-Access: read-write Status: current	`150`
cds3StatisticalAlarm Severity	`INTEGER {minor(1), major(2)}`	Sets up the severity of any of the statistical alarms. When a statistical counter exceeds its specified threshold, the system sends the alarm with the appropriate severity. Max-Access: read-write Status: current	`Minor`
cds3LCV15Min Threshold	`Unsigned32 (0..2147483647)`	Determines the value set in this object.When the threshold exceeds a 15-minute window, the system declares a statistical alarm. Refer to RFC 2496. Max-Access: read-write Status: current	none
cds3LCV24Hr Threshold	`Unsigned32 (0..2147483647)`	Determines the value set in this object. When the threshold exceeds a sliding 24-hour window, the system declares a statistical alarm. The threshold is checked every 15-minutes for the preceding 24-hour interval. Refer to RFC 2496. Max-Access: read-write Status: current	none
cds3LES15Min Threshold	`Unsigned32 (0..2147483647)`	Determines the value set in this object. When the threshold exceeds a 15-minute window, the system declares a statistical alarm. Refer to RFC 2496. Max-Access: read-write Status: current	none

Table 3-22 DS3 Alarm Configuration Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cds3LES24HrThreshold	Unsigned32 (0..2147483647)	Determines the value set in this object. When the threshold exceeds a sliding 24-hour window, the system declares a statistical alarm. The threshold is checked every 15 minutes for the preceding 24-hour interval. Refer to RFC 2496. Max-Access: read-write Status: current	none
cds3PCV15Min Threshold	Unsigned32 (0..2147483647)	Determines the value set in this object. When the threshold exceeds a 15-minute window, the system declares a statistical alarm. **Note** This object does not apply to the E3 interface. Refer to RFC 2496. Max-Access: read-write Status: current	none
cds3PCV24HrThreshold	Unsigned32 (0..2147483647)	Determines the value set in this object. When the threshold exceeds a sliding 24-hour window, the system declares a statistical alarm. The threshold is checked every 15 minutes for the preceding 24-hour interval. **Note** This object does not apply to the E3 interface. Refer to RFC 2496. Max-Access: read-write Status: current	none
cds3PES15Min Threshold	Unsigned32 (0..2147483647)	Determines the value set in this object. When the threshold exceeds a 15-minute window, the system declares a statistical alarm. **Note** This object does not apply to the E3 interface. Refer to RFC 2496. Max-Access: read-write Status: current	none
cds3PES24HrThreshold	Unsigned32 (0..2147483647)	Determines the value set in this object. When the threshold exceeds a sliding 24-hour window, the system declares a statistical alarm. The threshold is checked every 15 minutes for the preceding 24-hour interval. **Note** This object does not apply to the E3 interface. Refer to RFC 2496. Max-Access: read-write Status: current	none

Table 3-22 DS3 Alarm Configuration Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cds3PSES15Min Threshold	Unsigned32 (0..2147483647)	Determines the value set in this object. When the threshold exceeds a 15-minute window, the system declares a statistical alarm. **Note** This object does not apply to the E3 interface. Refer to RFC 2496. Max-Access: read-write Status: current	none
cds3PSES24Hr Threshold	Unsigned32 (0..2147483647)	Determines the value set in this object. When the threshold exceeds a sliding 24-hour window, the system declares a statistical alarm. The threshold is checked every 15 minutes for the preceding 24-hour interval. **Note** This object does not apply to the E3 interface. Refer to RFC 2496. Max-Access: read-write Status: current	none
cds3SEFS15Min Threshold	Unsigned32 (0..2147483647)	Determines the value set in this object. When the value exceeds a 15-minute window, the system declares a statistical alarm. Refer to RFC 2496. Max-Access: read-write Status: current	none
cds3SEFS24Hr Threshold	Unsigned32 (0..2147483647)	Determines the value set in this object. When the threshold exceeds a sliding 24-hour window, the system declares a statistical alarm. The threshold is checked every 15 minutes for the preceding 24-hour interval. Refer to RFC 2496. Max-Access: read-write Status: current	none
cds3UAS15Min Threshold	Unsigned32 (0..2147483647)	Determines the value set in this object. When the threshold exceeds a 15-minute window, the system declares a statistical alarm. Refer to RFC 2496. Max-Access: read-write Status: current	none

Table 3-22 DS3 Alarm Configuration Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cds3UAS24HrThreshold	Unsigned32 (0..2147483647)	Determines the value set in this object. When the threshold exceeds a sliding 24-hour window, the system declares a statistical alarm. The threshold is checked every 15 minutes for the preceding 24-hour interval. Refer to RFC 2496. Max-Access: read-write Status: current	none
cds3CCV15Min Threshold	Unsigned32 (0..2147483647)	Determines the value set in this object. When the threshold exceeds a 15-minute window, the system declares a statistical alarm. **Note** This object does not apply to the E3 interface. Refer to RFC 2496. Max-Access: read-write Status: current	none
cds3CCV24Hr Threshold	Unsigned32 (0..2147483647)	Determines the value set in this object. When the threshold exceeds a sliding 24-hour window, the system declares a statistical alarm. The threshold is checked every 15 minutes for the preceding 24-hour interval. **Note** This object does not apply to the E3 interface. Refer to RFC 2496. Max-Access: read-write Status: current	none
cds3CES15Min Threshold	Unsigned32 (0..2147483647)	Determines the value set in this object. When the threshold exceeds a 15-minute window, the system declares a statistical alarm. **Note** This object does not apply to the E3 interface. Refer to RFC 2496. Max-Access: read-write Status: current	none
cds3CES24HrThreshold	Unsigned32 (0..2147483647)	Determines the value set in this object. When the threshold exceeds a sliding 24-hour window, the system declares a statistical alarm. **Note** This object does not apply to the E3 interface. Refer to RFC 2496. Max-Access: read-write Status: current	none

Table 3-22 DS3 Alarm Configuration Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cds3CSES15Min Threshold	`Unsigned32 (0..2147483647)`	Determines the value set in this object. When the threshold exceeds a 15-minute window, the system declares a statistical alarm. **Note** This object does not apply to the E3 interface. Refer to RFC 2496. Max-Access: read-write Status: current	none
cds3CSES24Hr Threshold	`Unsigned32 (0..2147483647)`	Determines the value set in this object. When the threshold exceeds a sliding 24-hour window, the system declares a statistical alarm. **Note** This object does not apply to the E3 interface. Refer to RFC 2496. Max-Access: read-write Status: current	none

Table 3-22 DS3 Alarm Configuration Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cds3LineStatistical AlarmStateE	Unsigned32 (0..2147483647)	Specifies a bitmap of the DS3 Line Statistical alarms. 0 indicates no alarms. Max-Access: read-only Status: current The following bit positions itemize the individual alarm bits:	0

Bit Position	Alarm
1	PES current 15-minute threshold exceeded
2	PES 24-hour threshold exceeded
3	PSES current 15-minute threshold exceeded
4	PSES 24-hour threshold exceeded
5	SEFS current 15-minute threshold exceeded
6	SEFS 24-hour threshold exceeded
7	UAS current 15-minute threshold exceeded
8	UAS 24-hour threshold exceeded
9	LCV current 15-minute threshold exceeded
10	LCV 24-hour threshold exceeded
11	PCV current 15-minute threshold exceeded
12	PCV 24-hour threshold exceeded
13	LES current 15-minute threshold exceeded
14	LES 24-hour threshold exceeded
15	CCV current 15-minute threshold exceeded
16	CCV 24-hour threshold exceeded
17	CES current 15-minute threshold exceeded
18	CES 24-hour threshold exceeded
19	CSES current 15-minute threshold exceeded
20	CSES 24-hour threshold exceeded

DS3 Statistics Group

The object identifiers are listed in Table 3-23.

Table 3-23 DS3 Statistics Group Object Identifiers

Name	Object Identifier
cds3StatsTable	::= {cds3Stats 1}
cds3RcvLOSCount	::= {cds3StatsEntry 1}
cds3RcvOOFCount	::= {cds3StatsEntry 2}

Table 3-23 DS3 Statistics Group Object Identifiers (continued)

Name	Object Identifier
cds3RAICount	::= {cds3StatsEntry 3}
cds3CCVCount	::= {cds3StatsEntry 4}
cds3FECount	::= {cds3StatsEntry 5}
cds3RcvAISCount	::= { cds3StatsEntry 11 }

The MIB objects are listed in Table 3-24.

Table 3-24 DS3 Statistics Group MIB Objects

Name	Syntax	Description	Default Value
cds3StatsTable	`Sequence of cds3StatsEntry`	Describes the T3/E3 Statistics table. This table maintains the number of times the line encountered Loss of Signal (LOS), Out of Frame (OOF), Alarm Indication Signals(AIS), Remote Alarm Indications (RAI), C-bit Coding Violations (CCV), Framing Errors (FE), from the time it goes up. The line fails and goes down as a result of failures. When the line is brought back up by the user, the error statistics are cleared. Max-Access: not-accessible Status: current	none
cds3StatsEntry	`cds3StatsEntry`	Provides an entry in the T3/E3 Statistics Table. Statistics are gathered for each existing T3/E3 line. An entry is created automatically for each T3/E3 line that is activated. Max-Access: not-accessible Status: current	none
cds3RcvLOSCount	`Counter32`	Determines the number of times Loss of Signal (LOS) is detected with or without integrating to LOS alarm. Max-Access: read-only Status: current	none
cds3RcvOOFCount	`Counter32`	Determines the number of times OOF is detected with or without integrating to OOF alarm. Max-Access: read-only Status: current	none
cds3RAICount	`Counter32`	Determines the number of times RAI is detected with or without integrating to RAI alarm. Max-Access: read-only Status: current	none
cds3CCVCount	`Counter32`	Specifies the counter associated with the number of C-Bit Coding Violations encountered by a T3/E3 interface. Max-Access: read-only Status: current	none

Table 3-24 DS3 Statistics Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cds3FECount	Counter32	Determines the number of Framing Errors encountered by a T3/E3 interface. Max-Access: read-only Status: current	none
cds3RcvAISCount	Counter32	Determines the number of times Alarm Indication Signals (AIS) are detected. AIS indicates an upstream failure is detected by the far end. For more information, refer to RFC 2496, Section 2.4.3. Max-Access: read-only Status: current	none

DS3 PLCP Alarm Configuration Group

Table 3-25 lists the object identifiers used for the DS3 Physical Layer Convergence Procedure (PLCP) Alarm Configuration Group.

Table 3-25 DS3 PLCP Alarm Configuration Group Object Identifiers

Name	Object Identifier
cds3AlarmConfigPlcpTable	::= {cds3AlarmConfig 2}
cds3AlarmConfigPlcpEntry	::= {cds3AlarmConfigPlcpTable 1}
cds3PlcpStatisticalAlarmSeverity	::= {cds3AlarmConfigPlcpEntry 1}
cds3PlcpBip8CV15MinThreshold	::= {cds3AlarmConfigPlcpEntry 2}
cds3PlcpBip8CV24HrThreshold	::= {cds3AlarmConfigPlcpEntry 3}
cds3PlcpBip8ES15MinThreshold	::= {cds3AlarmConfigPlcpEntry 4}
cds3PlcpBip8ES24HrThreshold	::= {cds3AlarmConfigPlcpEntry 5}
cds3PlcpBip8SES15MinThreshold	::= {cds3AlarmConfigPlcpEntry 6}
cds3PlcpBip8SES24HrThreshold	::= {cds3AlarmConfigPlcpEntry 7}
cds3PlcpSEFS15MinThreshold	::= {cds3AlarmConfigPlcpEntry 8}
cds3PlcpSEFS24HrThreshold	::= {cds3AlarmConfigPlcpEntry 9}
cds3PlcpUAS15MinThreshold	::= {cds3AlarmConfigPlcpEntry 10}
cds3PlcpUAS24HrThreshold	::= {cds3AlarmConfigPlcpEntry 11}

The MIB objects are listed in Table 3-26.

Table 3-26 DS3 PLCP Alarm Configuration Group MIB Objects

Name	Syntax	Description	Default Value
cds3AlarmConfigPlcp Table	`Sequence of cds3AlarmConfigPlcp Entry`	Describes the ATM interface PLCP alarm configuration table. PLCP is a sublayer over the DS3 interface that carries ATM cells. Max-Access: not-accessible Status: current	none
cds3AlarmConfigPlcp Entry	`cds3AlarmConfigPlcp Entry`	Provides an entry in the DS3 PLCP Alarm Configuration table. The entries contain parameters to configure DS3 PLCP error statistics thresholds, so that an alarm is raised whenever any threshold is exceeded. The entries are automatically created for active DS3 interfaces with PLCP framing enabled, with default values for objects. Max-Access: not-accessible Status: current The index is `ifIndex`.	none
cds3PlcpStatistical AlarmSeverity	`INTEGER {minor(1), major(2)}`	Sets up the severity of PLCP statistical alarm. When a PLCP Statistical alarm is detected, the module can send the alarm with appropriate severity. Max-Access: read-write Status: current	none
cds3PlcpBip8CV15Min Threshold	`Unsigned32 (0..2147483647)`	Determines the value set in this object. If the threshold exceeds a 15-minute window, the system declares a statistical alarm. Max-Access: read-write Status: current	none
cds3PlcpBip8CV24Hr Threshold	`Unsigned32 (0..2147483647)`	Determines the value set in this object. If the threshold exceeds a sliding 24-hour window, the system declares a statistical alarm. The threshold is checked every 15 minutes for the preceding 24-hour interval. Max-Access: read-write Status: current	none
cds3PlcpBip8ES15Min Threshold	`Unsigned32 (0..2147483647)`	Determines the value set in this object. When the threshold exceeds a 15-minute window, the system declares a statistical alarm. Max-Access: read-write Status: current	none

Table 3-26 DS3 PLCP Alarm Configuration Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cds3PlcpBip8ES24Hr Threshold	Unsigned32 (0..2147483647)	Determines the value set in this object. When the threshold exceeds a sliding 24-hour window, the system declares a statistical alarm. The threshold is checked every 15 minutes for the preceding 24-hour interval. Max-Access: read-write Status: current	none
cds3PlcpBip8SES15Min Threshold	Unsigned32 (0..2147483647)	Determines the value set in this object. When the threshold exceeds a 15-minute window, the system declares a statistical alarm. Max-Access: read-write Status: current	none
cds3PlcpBip8SES24Hr Threshold	Unsigned32 (0..2147483647)	Determines the value set in this object. When the threshold exceeds a sliding 24-hour window, the system declares a statistical alarm. The threshold is checked every 15 minutes for the preceding 24-hour interval. Max-Access: read-write Status: current	none
cds3PlcpSEFS15Min Threshold	Unsigned32 (0..2147483647)	Determines the value set in this object. When the threshold exceeds a 15-minute window, the system declares a statistical alarm. Max-Access: read-write Status: current	none
cds3PlcpSEFS24Hr Threshold	Unsigned32 (0..2147483647)	Determines the value set in this object. When the threshold exceeds a sliding 24-hour window, the system declares a statistical alarm. The threshold is checked every 15 minutes for the preceding 24-hour interval. Max-Access: read-write Status: current	none
cds3PlcpUAS15Min Threshold	Unsigned32 (0..2147483647)	Determines the value set in this object. When the threshold exceeds a 15-minute window, the system declares a statistical alarm. Max-Access: read-write Status: current	none
cds3PlcpUAS24Hr Threshold	Unsigned32 (0..2147483647)	Determines the value set in this object. When the threshold exceeds a sliding 24-hour window, the system declares a statistical alarm. The threshold is checked every 15 minutes for the preceding 24-hour interval. Max-Access: read-write Status: current	none

DS3 PLCP Alarm Table

The object identifiers are listed in Table 3-27.

Table 3-27 DS3 PLCP Alarm Table Object Identifiers

Name	Object Identifier
cds3AlarmPlcpTable	::= {cds3Alarm 2}
cds3AlarmPlcpEntry	::= {cds3AlarmPlcpTable 1}
cds3PlcpLineAlarmState	::= {cds3AlarmPlcpEntry 1}
cds3PlcpLineStatisticalAlarmState	::= {cds3AlarmPlcpEntry 2}
cds3PlcpBip8CVCurrent	::= {cds3AlarmPlcpEntry 3}
cds3PlcpBip8CV24HrBucket	::= {cds3AlarmPlcpEntry 4}
cds3PlcpBip8ESCurrent	::= {cds3AlarmPlcpEntry 5}
cds3PlcpBip8ES24HrBucket	::= {cds3AlarmPlcpEntry 6}
cds3PlcpBip8SESCurrent	::= {cds3AlarmPlcpEntry 7}
cds3PlcpBip8SES24HrBucket	::= {cds3AlarmPlcpEntry 8}
cds3PlcpSEFSCurrent	::= {cds3AlarmPlcpEntry 9}
cds3PlcpSEFS24HrBucket	::= {cds3AlarmPlcpEntry 10}
cds3PlcpUASCurrent	::= {cds3AlarmPlcpEntry 11}
cds3PlcpUAS24HrBucket	::= {cds3AlarmPlcpEntry 12}

The MIB objects are listed in Table 3-28.

Table 3-28 DS3 PLCP Alarm Table MIB Objects

Name	Syntax	Description	Default Value
cds3AlarmPlcpTable	`Sequence of cds3AlarmPlcpEntry`	Describes the PLCP interface alarm table. This table maintains the CV, ES, SES, SEFS, and UAS for DS3 line with PLCP framing selected. Refer to RFC 2496 for a description of these error statistics. Max-Access: not-accessible Status: current	none
cds3AlarmPlcpEntry	`Cds3AlarmPlcpEntry`	Provides an entry in the DS3 PLCP alarm table. Bucket statistics (15-minute and 24-hour interval statistics) are gathered for each existing DS3 line with PLCP frame format selected. An entry is created automatically and is indexed by the DS3 line number. Max-Access: not-accessible Status: current The index is `ifIndex`.	none

Table 3-28 DS3 PLCP Alarm Table MIB Objects (continued)

Name	Syntax	Description		Default Value
cds3PlcpLineAlarm State	`BITS {rxRAI(0), txRAI(1), rcvOOF(2), localLpbk(3), remoteLpbk(4)}`	Specifies a bitmap of the DS3 Line Alarms. A value of 0 for this object indicates no alarms. Max-Access: read-only Status: current The bit positions and alarms include:		none
		Bit Position	**Alarm**	
		0	Receiving RAI	
		1	Transmitting RAI	
		2	Receiving OOF	
		3	Near End Local Loopback in effect	
		4	Near End Remote Loopback in effect	
cds3PlcpLine StatisticalAlarmState	`BITS {cds3PlcpBip8CVCurrent(0), cds3PlcpBip8CV24HrBucket(1), cds3PlcpBip8ESCurrent(2), cds3PlcpBip8ES24HrBucket(3), cds3PlcpBip8SESCurrent(4), cds3PlcpBip8SES24HrBucket(5), cds3PlcpSEFSCurrent(6), cds3PlcpSEFS24HrBucket(7), cds3PlcpUASCurrent(8), cds3PlcpUAS24HrBucket(9)}`	Specifies a bitmap of the DS3 PLCP Line Statistical Alarms. A value of 0 for this object indicates no alarms. Max-Access: read-only Status: current The bit positions and alarms include:		none
		Bit Position	**Alarm**	
		0	Bip8CV 15-minute threshold	
		1	Bip8CV 24-hour threshold	
		2	Bip8ES 15-minute threshold	
		3	Bip8ES 24-hour threshold	
		4	Bip8SES 15-minute threshold	
		5	Bip8SES 24-hour threshold	
		6	Plcp SEFS 15-minute threshold	
		7	Plcp SEFS 24-hour threshold	
		8	Plcp UAS 15-minute threshold	
		9	Plcp UAS 24-hour threshold	
cds3PlcpBip8CV Current	`Counter32`	Determines the number of BIP-8 code violations (CV) encountered by the line in the current 15-minute period. Max-Access: read-only Status: current		none
cds3PlcpBip8CV24 HrBucket	`Counter32`	Determines the number of BIP-8 CVs encountered by the line in the last 24 hours. Max-Access: read-only Status: current		none

Table 3-28 DS3 PLCP Alarm Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cds3PlcpBip8ES Current	Counter32	Determines the number of BIP-8 Errored Seconds (ES) encountered by the line in the current 15-minute period. Max-Access: read-only Status: current	none
cds3PlcpBip8ES24Hr Bucket	Counter32	Determines the number of BIP-8 errored seconds encountered by the line in the last 24 hours. Max-Access: read-only Status: current	none
cds3PlcpBip8SES Current	Counter32	Determines the number of BIP-8 Severely Errored Seconds (SES) encountered by the line in the current 15-minute period. Max-Access: read-only Status: current	none
cds3PlcpBip8SES24 HrBucket	Counter32	Determines the number of BIP-8 SES encountered by the line in the last 24 hours. Max-Access: read-only Status: current	none
cds3PlcpSEFS Current	Counter32	Determines the number of BIP-8 Severely Errored Framing Seconds (SEFS) encountered by the line in the current 15-minute period. Max-Access: read-only Status: current	none
cds3PlcpSEFS24Hr Bucket	Counter32	Determines the number of BIP-8 SEFS encountered by the line in the last 24 hours. Max-Access: read-only Status: current	none
cds3PlcpUASCurrent	Counter32	Determines the number of Unavailable Seconds (UAS) encountered by the line in the current 15-minute period. Max-Access: read-only Status: current	none
cds3PlcpUAS24Hr Bucket	Counter32	Determines the number of BIP-8 UAS encountered by the line in the last 24 hours. Max-Access: read-only Status: current	none

DS3 PLCP Statistics Group

The object identifiers are listed in Table 3-29.

Table 3-29 DS3 PLCP Statistics Group Object Identifiers

Name	Object Identifier
cds3PlcpStatsTable	::= {cds3Stats 2}
cds3PlcpStatsEntry	::= {cds3PlcpStatsTable 1}
cds3PlcpRcvBip8Count	::= {cds3PlcpStatsEntry 1}
cds3PlcpRcvOOFCount	::= {cds3PlcpStatsEntry 2}
cds3PlcpRcvRAICount	::= {cds3PlcpStatsEntry 3}
cds3PlcpFECount	::= {cds3PlcpStatsEntry 4}
cds3PlcpFESecCount	::= {cds3PlcpStatsEntry 5}
cds3PlcpSEFSecCountE	::= {cds3PlcpStatsEntry 6}
cds3PlcpFEBECount	::= {cds3PlcpStatsEntry 7}
cds3PlcpFEBESecCount	::= {cds3PlcpStatsEntry 8}
cds3PlcpSEFEBESecCount	::= {cds3PlcpStatsEntry 9}

A list of the MIB objects is shown in Table 3-30.

Table 3-30 DS3 PLCP Statistics Group MIB Objects

Name	Syntax	Description	Default Value
cds3PlcpStatsTable	`Sequence of cds3PlcpStatsEntry`	Describes the 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 as a result of failures. When the line is brought back up again by the user after eliminating the error conditions, the statistics are cleared. Max-Access: not-accessible Status: current	none
cds3PlcpStatsEntry	`cds3PlcpStatsEntry`	Contains an entry in the T3 PLCP Statistics table. Statistics are gathered for each existing T3 line with PLCP frame format selected. An entry is created automatically and is indexed by the T3 line number. Max-Access: not-accessible Status: current The index is `ifIndex`.	none
cds3PlcpRcvBip8Count	`Counter32`	Determines the number of BIP-8 errors encountered by the line. Max-Access: read-only Status: current	none

Table 3-30 DS3 PLCP Statistics Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cds3PlcpRcvOOFCount	Counter32	Determines the number of times OOF is detected with or without integrating to OOF alarm. Max-Access: read-only Status: current	none
cds3PlcpRcvRAICount	Counter32	Determines the number of times RAI is detected with or without integrating to RAI alarm. Max-Access: read-only Status: current	none
cds3PlcpFECount	Counter32	Detects the number of Framing Errors. Max-Access: read-only Status: current	none
cds3PlcpFESecCount	Counter32	Detects the number of Framing Error ES. Max-Access: read-only Status: current	none
cds3PlcpSEFSecCountE	Counter32	Detects the number of SEFES. Max-Access: read-only Status: current	none
cds3PlcpFEBECount	Counter32	Detects the count of FEBE. Max-Access: read-only Status: current	none
cds3PlcpFEBESecCount	Counter32	Detects the number of FEBE ES. Max-Access: read-only Status: current	none
cds3PlcpSEFEBESec Count	Counter32	Detects the number of SEFS. Max-Access: read-only Status: current	none

Previous 24 Hour Interval Data

The object identifiers are listed in Table 3-31.

Table 3-31 Previous 24 Hour Interval Data Object Identifiers

Name	Object Identifier
cds3Previous24HrTable	::= { cds3Stats 6 }
cds3Previous24HrEntry	::= { cds3Previous24HrTable 1 }
cds3LCVPrevious24Hr	::= { cds3Previous24HrEntry 1 }

Table 3-31 Previous 24 Hour Interval Data Object Identifiers (continued)

Name	Object Identifier
cds3LESPrevious24Hr	::= { cds3Previous24HrEntry 2 }
cds3PCVPrevious24Hr	::= { cds3Previous24HrEntry 3 }
cds3PESPrevious24Hr	::= { cds3Previous24HrEntry 4 }
cds3PSESPrevious24Hr	::= { cds3Previous24HrEntry 5 }
cds3SEFSPrevious24Hr	::= { cds3Previous24HrEntry 6 }
cds3UASPrevious24Hr	::= { cds3Previous24HrEntry 7 }
cds3CCVPrevious24Hr	::= { cds3Previous24HrEntry 8 }
cds3CESPrevious24Hr	::= { cds3Previous24HrEntry 9 }
cds3CSESPrevious24Hr	::= { cds3Previous24HrEntry 10 }

A list of the MIB objects is shown in Table 3-32.

Table 3-32 Previous 24 Hour Interval Data MIB Objects

Name	Syntax	Description	Default Value
cds3Previous24HrTable	`SEQUENCE OF Cds3Previous24HrEntry`	Describes the DS3 interface previous 24 hour table. The `cds3Previous24HrTable` table contains counters for the previous 24 hour interval. **Note** This table is optional. Max-Access: not-accessible Status: current	none
cds3Previous24HrEntry	`Cds3Previous24HrEntry`	Determines an entry in the T3/E3 current 24-hour counter table. When interface is made active, the entries are automatically created for an `ifType` value of `ds3(30)`. Max-Access: not-accessible Status: current The index is `ifIndex`.	none
cds3LCVPrevious24Hr	`Counter32`	Determines the number of LCV encountered by the line in the previous 24 hour period. Max-Access: read-only Status: current	none
cds3LESPrevious24Hr	`Counter32`	Determines the number of LAN Emulation Server (LES) encountered by the line in the previous 24-hour period. Max-Access: read-only Status: current	none

Table 3-32 Previous 24 Hour Interval Data MIB Objects (continued)

Name	Syntax	Description	Default Value
cds3PCVPrevious24Hr	Counter32	Determines the number of PCVs encountered by the line in the previous 24-hour period. Max-Access: read-only Status: current	none
cds3PESPrevious24Hr	Counter32	Determines the number of PES encountered by the line in the previous 24-hour period. Max-Access: read-only Status: current	none
cds3PSESPrevious24Hr	Counter32	Determines the number of PSESs encountered by the line in the previous 24-hour period. Max-Access: read-only Status: current	none
cds3SEFSPrevious24Hr	Counter32	Determines the number of SEFSs encountered by the line during the start of the current 24-hour period. Max-Access: read-only Status: current	none
cds3UASPrevious24Hr	Counter32	Determines the number of UAS encountered by the line in the previous 24-hour period. Max-Access: read-only Status: current	none
cds3CCVPrevious24Hr	Counter32	Determines the number of CCVs encountered by the line in the previous 24-hour period. Max-Access: read-only Status: current	none
cds3CESPrevious24Hr	Counter32	Determines the number of Circuit Emulation Service (CES) encountered by the line in the previous 24 hour period. Max-Access: read-only Status: current	none
cds3CSESPrevious24Hr	Counter32	Determines the number of CSES encountered by the line in the previous 24-hour period. Max-Access: read-only Status: current	none

Cisco DS3 Units of Conformance

The object identifiers are listed in Table 3-33.

Table 3-33 Cisco DS3 MIB Units of Conformance Object Identifiers

Name	Object Identifier
ciscoDs3ConfMIBGroup	::= {ciscoDs3MIBGroups 1}
cds3StatsMIBGroup	::= {ciscoDs3MIBGroups 2}
ciscoDs3AlarmMIBGroup	::= {ciscoDs3MIBGroups 3}
ciscoPlcpCounterMIBGroup	::= {ciscoDs3MIBGroups 4}
ciscoPlcpAlarmMIBGroup	::= {ciscoDs3MIBGroups 5}

A list of the MIB objects is listed in Table 3-34.

Table 3-34 Cisco DS3 MIB Units of Conformance MIB Objects

Name	Objects	Description	Default Value
ciscoDs3ConfMIBGroup	`cds3LineType, cds3LineAIScBitsCheck, cds3LineRcvFEACValidation, cds3LineOOFCriteria, cds3TraceToTransmit, cds3TraceToExpect, cds3TraceAlarm`	Relates to the `cds3Config` group. Status: current	none
cds3StatsMIBGroup	`cds3RcvLOSCount, cds3RcvOOFCount, cds3RAICount, cds3CCVCount, cds3FECount`	Relates to the `cds3Stats` group. Status: current	none

Table 3-34 Cisco DS3 MIB Units of Conformance MIB Objects (continued)

Name	Objects	Description	Default Value
ciscoDs3AlarmMIBGroup	cds3NEAlarmUpCount, cds3NEAlarmDownCount, cds3NEAlarmThreshold, cds3FEAlarmUpCount, cds3FEAlarmDownCount, cds3FEAlarmThreshold, cds3StatisticalAlarmSeverity, cds3LCV15MinThreshold, cds3LCV24HrThreshold, cds3LES15MinThreshold, cds3LES24HrThreshold, cds3PCV15MinThreshold, cds3PCV24HrThreshold, cds3PES15MinThreshold, cds3PES24HrThreshold, cds3PSES15MinThreshold, cds3PSES24HrThreshold, cds3SEFS15MinThreshold, cds3SEFS24HrThreshold, cds3UAS15MinThreshold, cds3UAS24HrThreshold, cds3CCV15MinThreshold, cds3CCV24HrThreshold, cds3CES15MinThreshold, cds3CES24HrThreshold, cds3CSES15MinThreshold, cds3CSES24HrThreshold, cds3LineStatisticalAlarmState	Relates to the cds3AlarmConfig group. Status: current	none

Table 3-34 Cisco DS3 MIB Units of Conformance MIB Objects (continued)

Name	Objects	Description	Default Value
ciscoPlcpCounterMIBGroup	cds3PlcpRcvBip8Count, cds3PlcpRcvOOFCount, cds3PlcpRcvRAICount, cds3PlcpFECount, cds3PlcpFESecCount, cds3PlcpSEFSecCount, cds3PlcpFEBECount, cds3PlcpFEBESecCount, cds3PlcpSEFEBESecCount	Relates to the cds3PlcpStats group. Status: current	none
ciscoPlcpAlarmMIBGroup	cds3PlcpStatisticalAlarmSeverity, cds3PlcpBip8CV15MinThreshold, cds3PlcpBip8CV24HrThreshold, cds3PlcpBip8ES15MinThreshold, cds3PlcpBip8ES24HrThreshold, cds3PlcpBip8SES15MinThreshold, cds3PlcpBip8SES24HrThreshold, cds3PlcpSEFS15MinThreshold, cds3PlcpSEFS24HrThreshold, cds3PlcpUAS15MinThreshold, cds3PlcpUAS24HrThreshold, cds3PlcpLineAlarmState, cds3PlcpLineStatisticalAlarmState, cds3PlcpBip8CVCurrent, cds3PlcpBip8CV24HrBucket, cds3PlcpBip8ESCurrent, cds3PlcpBip8ES24HrBucket, cds3PlcpBip8SESCurrent, cds3PlcpBip8SES24HrBucket, cds3PlcpSEFSCurrent, cds3PlcpSEFS24HrBucket, cds3PlcpUASCurrent, cds3PlcpUAS24HrBucket	Relates to the Cds3AlarmPlcp group. Status: current	none

Cisco SONET

This section describes the Cisco SONET MIB module, which resides in the CISCO-SONET-MIB.my file, describes SONET/SDH interface objects. It is an extension to the standard SONET MIB. Refer to RFC 2558.

Note The Cisco SONET MIB is supported by AXSM and AXSM-E.

The MIB objects used for Cisco SONET include:

- SONET Configuration Table
- SONET APS Configuration Table
- SONET Section Total Table
- SONET Section Trace Table
- SONET Line Total Table
- SONET Far End Line Total Table

- SONET Path Total Table
- SONET Far End Path Total Table
- Path Trace Table
- SONET Statistics Tables
- Cisco SONET Conformance Information

SONET Configuration Table

The object identifiers are listed in Table 3-35.

Table 3-35 SONET Configuration Table Object Identifiers

Name	Object Identifier
csConfigTable	::= {csConfig 1}
csConfigEntry	::= {csConfigTable 1}
csConfigLoopbackType	::= {csConfigEntry 1}
csConfigXmtClockSource	::= {csConfigEntry 2}
csConfigFrameScramble	::= {csConfigEntry 3}
csConfigType	::= {csConfigEntry 4}

The MIB objects are listed in Table 3-36.

Table 3-36 SONET Configuration Table MIB Objects

Name	Syntax	Description	Default Value
csConfigTable	`Sequence of csConfigEntry`	Describes the SONET/SDH configuration table. This table has objects for configuring SONET lines. Max-Access: not-accessible Status: current	none
csConfigEntry	`csConfigEntry`	Provides an entry in the table. There is an entry for each SONET line in the table. Entries are automatically created for an `ifType` value of `SONET(39)`. The object `ifAdminStatus` from the `ifTable` must be used to enable or disable a line. By default, the line state is down. To maintain the integrity of the reference, the following is required: • The value of `ifAdminStatus` from `ifTable` for entries in this table cannot be set to `down (2)` if there are virtual interfaces configured on the line. • The value of `csConfigLoopbackType` must be set to `noLoopback (1)`, before any loopback type is changed from one mode to another. Max-Access: not-accessible Status: current Created By: Agent The index is `ifIndex`.	none
csConfigLoopback Type	`INTEGER {noLoopback (1), lineLocal (2), lineRemote (3)}`	Represents the desired loopback configuration of the SONET line. Max-Access: read-write Status: current The following are the loopback configuration choices:	`noLoopback`

Loopback Configuration	Description
`noLoopback`	Not in the loopback state.
`lineLocal`	The signal sent on this line is looped back through the device.
`lineRemote`	The sent signal does not go through the device but is looped back out.

Table 3-36 SONET Configuration Table MIB Objects (continued)

Name	Syntax	Description		Default Value
csConfigXmtClock Source	INTEGER {loopTiming (1), localTiming (2)}	Defines the source of the transmit clock. Max-Access: read-write Status: current The following are the clock source choices:		localTiming
		Clock Source	**Description**	
		loopTiming	Indicates that the recovered receive clock is used as the transmit clock.	
		localTiming	Indicates that a local clock source is used or that an external clock is attached to the box containing the interface.	
csConfigFrame Scramble	INTEGER {disabled(1), enabled(2)}	Disables or enables the Scrambling option in the SONET line. Refer to Bellcore GR-253-CORE for frame scrambling. Max-Access: read-write Status: current		enabled
csConfigType	INTEGER {SONETSts3c(1), SONETStm1(2), SONETSts12c(3), SONETStm4(4), SONETSts48c(5), SONETStm16(6), sonetSts192c(7), sonetStm64(8)}	Represents the configured line type. This object is used to switch between SONET (North American) and SDH (European) format. Sts is SONET format, while Stm is SDH format. Max-Access: read-only Status: current The following are the configured line types and applications:		none
		Configured Line Type	**Application**	
		SONETSts3c	OC3	
		SONETStm1	European standard OC3	
		SONETSts12c	OC12	
		SONETStm4	European standard OC12	
		SONETSts48c	OC48	
		SONETStm16	European standard OC48	
		SONETSts192c	OC-192	
		SONETStm64	European standard OC-192	

SONET APS Configuration Table

The object identifier for each MIB object for the SONET Automatic Protection Switching (APS) Configuration Table is listed in Table 3-37.

Table 3-37 SONET APS Configuration Table Object Identifiers

Name	Object Identifier
csApsConfigTable	::= {csApsConfig 1}
csApsConfigEntry	::= {csApsConfigTable 1}
csApsWorkingIndex	::= {csApsConfigEntry 1}
csApsProtectionIndex	::= {csApsConfigEntry 2}
csApsEnable	::= {csApsConfigEntry 3}
csApsArchMode	::= {csApsConfigEntry 4}
csApsActiveLine	::= {csApsConfigEntry 5}
csApsSigFaultBER	::= {csApsConfigEntry 6}
csApsSigDegradeBER	::= {csApsConfigEntry 7}
csApsWaitToRestore	::= {csApsConfigEntry 8}
csApsDirection	::= {csApsConfigEntry 9}
csApsRevertive	::= {csApsConfigEntry 10}
csApsDirectionOperational	::= {csApsConfigEntry 11}
csApsArchModeOperational	::= {csApsConfigEntry 12}
csApsChannelProtocol	::= {csApsConfigEntry 13}
csApsLineFailureCode	::= {csApsConfig 2}
csApsLineSwitchReason	::= {csApsConfig 3}

The MIB objects are listed in Table 3-38.

Table 3-38 SONET APS Configuration Table MIB Objects

Name	Syntax	Description	Default Value
csApsConfig Table	`Sequence of CsApsConfigEntry`	Contains objects to configure the APS feature in a SONET Line. APS has the ability to configure a pair of SONET lines for redundancy. When the active line fails, the hardware will automatically switch the active line from the working line to the protection line or vice versa, within 60 m. Max-Access: not-accessible Status: current	none
csApsConfig Entry	`csApsConfigEntry`	Creates an entry when an APS pair is configured. To create an entry, the following objects must be specified: • `csApsWorkingIndex` • `csApsProtectionIndex` • `csApsEnable` • `csApsArchMode` The protection line must not be active. For example, `ifAdminStatus` must be down, while configuring APS. An entry is created by setting the value of `csApsEnable` to `csApsEnabled (2)` and deleted by setting it to `csApsDisabled (1)`. Once a line is configured as working line or protection line, it remains in that role until APS is disabled on that SONET line pair. It remains in the working/protection role even after the card is reset. Max-Access: not-accessible Status: current The index is `csApsWorkingIndex`.	none

Table 3-38 SONET APS Configuration Table MIB Objects (continued)

<table>
<tr><th>Name</th><th>Syntax</th><th>Description</th><th>Default Value</th></tr>
<tr><td>csApsWorking Index</td><td>InterfaceIndex</td><td>Refers to the working line in the APS pair.
When a pair of APS lines is configured, one line must be the working line, which is the primary line, and the other must be the protection line, which is the backup line.
Max-Access: not-accessible
Status: current
The Chassis/Slot (CS) Format that is used for this index entry are as follows:
<table>
<tr><th>Field</th><th>Description</th><th>Bit Positions</th></tr>
<tr><td>Chassis #</td><td>Shelf number of the module that is plugged in.</td><td>24 to 31 (eight bits).</td></tr>
<tr><td>Logical slot #</td><td>Logical slot number of the module that is plugged in.</td><td>16 to 23 (eight bits).</td></tr>
<tr><td>Value</td><td>Refer to the description field for the MIB object.</td><td>0 to 15 (Refers to a number from 1 to 60).</td></tr>
</table></td><td>none</td></tr>
<tr><td>csApsProtection Index</td><td>InterfaceIndex</td><td>Indicates that the protection line becomes the active line when an APS switch occurs (an APS switch can occur because of a failure on the working line).
Max-Access: read-write
Status: current
The CS Format that is used for this index entry are as follows:
<table>
<tr><th>Field</th><th>Description</th><th>Bit Positions</th></tr>
<tr><td>Chassis #</td><td>Shelf Number of the module that is plugged in.</td><td>24 to 31 (eight bits).</td></tr>
<tr><td>Logical Slot #</td><td>Logical Slot Number of the module that is plugged in.</td><td>16 to 23 (eight bits).</td></tr>
<tr><td>Value</td><td>Refer to the description field for the MIB object.</td><td>0 to 15 (refers to a number from 1 to 60).</td></tr>
</table></td><td>none</td></tr>
<tr><td>csApsEnable</td><td>INTEGER {csApsDisabled (1), csApsEnabled (2)}</td><td>Enables or disables the APS feature on the working/protection line pairs. When enabled, the hardware will automatically switch the active line from the working line to the protection line within 60 meters, or vice versa.
Max-Access: read-write
Status: current</td><td>none</td></tr>
</table>

Table 3-38 SONET APS Configuration Table MIB Objects (continued)

Name	Syntax	Description	Default Value
csApsArchMode	INTEGER {onePlusOne(1), oneToOne(2), anexBOnePlusOne(3), ycableOnePlusOneNok1k2(4), straightOnePlusOneNok1k2(5)}	Configures the APS architecture mode on the working/protection line pairs. See csApsK1K2Disable for description of K1K2 bytes. All of the following are supported on single slot. oneToOne(2) is not supported across 2 slots; which is the working and protection slot numbers must be the same in oneToOne(2). • onePlusOne—Supports the same card used across two cards. This mode means that the transmit and receive signals go only over the active line (which could be working or protection line). (Straight cable implied.) • oneToOne—Supports only the same card. This mode means that the transmit and receive signals go over the working and protection lines. (Straight cable implied.) • anexBOnePlusOne—Supports the same card used across two cards. This mode is like the onePlusOne mode, except that the csApsDirection can only be bi-directional. (Straight cable implied.) • ycableOnePlusOneNok1k2—With Y-cable ignore K1K2 bytes. This mode is the Y-cable redundancy mode. • straightOnePlusOneNok1k2—With straight cable, ignore K1K2 bytes. This mode is like onePlusOne, but with K1, K2 bytes are ignored. Max-Access: read-write Status: current	none
csApsActiveLine	INTEGER {csApsWorkingLine(1), csApsProtectionLine(2), csApsNone(3)}	Indicates which line is currently active. It is the working line, protection line, or none if neither lines are active. This object reflects the status of the receive direction. Max-Access: read-only Status: current	none
csApsSigFault BER	Unsigned32 (3..5)	Contains the Bit Error Rate (BER) threshold for Signal Fault detection on the working line. Once this threshold is exceeded, an APS switch can occur. This value is 10 to -*n*. For example, *n* is between 3 and 5. Max-Access: read-write Status: current	3

Table 3-38 SONET APS Configuration Table MIB Objects (continued)

Name	Syntax	Description	Default Value
csApsSigDegrade BER	`Unsigned32 (5..9)`	Contains the BER threshold for Signal Degrade detection on the working line. Once this threshold is exceeded, an APS switch can occur. This value is 10 to -*n*. For example, *n* is between 5 and 9. Max-Access: read-write Status: current	`5`
csApsWaitTo Restore	`Unsigned32 (1..12) in units of minutes`	Contains interval in minutes before attempting to switch back to the working line. It is not applicable if the line is configured in nonrevertive mode. The protection line continues to be active, even if failures on the working line are cleared. The framer clears the signal-fault and signal-degrade when APS switch occurs. Refer to `csApsRevertive` in this table for a description of nonrevertive. Max-Access: read-write Status: current	none
csApsDirection	`INTEGER {uniDirectional(1), biDirectional(2)}`	Configures the switching direction that this APS line supports. The direction is either unidirectional, where the APS switches are only in one direction, or bidirectional, where the APS switches both ends of the line. Max-Access: read-write Status: current	none
csApsRevertive	`INTEGER {nonrevertive(1), revertive(2)}`	Configures the APS revertive or nonrevertive option. Revertive will switch the working line back to active state after the Wait-To-Restore interval has expired and the working line Signal-Fault/Signal-Degrade has been cleared. See `csApsWaitToRestore` for a description of the Wait-To-Restore interval. With the nonrevertive option, the protection line continues to be the active line. The active line does not switch to the working line. Max-Access: read-write Status: current	none
csApsDirection Operational	`Integer {uniDirectional(1), biDirectional(2)}`	Shows the actual APS direction that is implemented on the Near End terminal. APS direction configured through `csApsDirection` is negotiated with the Far End and APS direction setting acceptable to both ends is operational at the Near End. Max-Access: read-only Status: current	none

Table 3-38 SONET APS Configuration Table MIB Objects (continued)

<table>
<tr><th>Name</th><th>Syntax</th><th>Description</th><th>Default Value</th></tr>
<tr><td>csApsArchMode Operational</td><td>Integer{onePlusOne(1), oneToOne(2), anexBOnePlusOne(3), ycableOnePlusOneNok1k2(4), straightOnePlusOneNok1k2(5)}</td><td>Shows the actual APS architecture mode that is implemented on the Near End terminal. APS architecture mode configured through csApsArchMode object is negotiated with the Far End through APS channel. Architecture mode acceptable to both the Near End and the Far End terminals is then operational at the Near End. This value can be different than the APS architecture mode configured.
Max-Access: read-only
Status: current</td><td>none</td></tr>
<tr><td>csApsChannel Protocol</td><td>Integer {bellcore(1), itu(2)}</td><td>Configures the APS channel protocol to be implemented at the Near End terminal. K1 and K2 overhead bytes in a SONET signal are used as an APS channel. This channel is used to carry APS protocol.
Max-Access: read-write
Status: current
The values include:
APS Channel Protocol — Defined In
bellcore(1) — Implements APS channel protocol as defined in Bellcore GR-253-CORE.
itu(2) — Implements APS channel protocol as defined in ITU document G.783, Annex A.</td><td>none</td></tr>
<tr><td>csApsLine FailureCode</td><td>INTEGER {csApsChannelMismatch(1), csApsProtectionByteFail(2), csApsFEProtectionFailure(3)}</td><td>Specifies the SONET APS line failure code, or the failure encountered by the APS line.
The failure descriptions include:
• csApsChannelMismatch—Indicates that the transmitted K1 byte and the received K2 byte do not match.
• csApsProtectionByteFail—Indicates that either a K1 byte with an invalid type of switch request bits was received, or the priority of the received K1 byte is lower than the transmitted K1 byte.
• csApsFEProtectionFailure—Indicates that a remote end error is detected.
Refer to Bellcore GR-253-CORE.
Max-Access: read-only
Status: current</td><td>none</td></tr>
</table>

Table 3-38 SONET APS Configuration Table MIB Objects (continued)

<table>
<tr><th>Name</th><th>Syntax</th><th>Description</th><th>Default Value</th></tr>
<tr><td>csApsLineSwitch Reason</td><td>INTEGER {csApsOther(1), csApsRevertive(2), csApsManual(3), csApsSignalDefectLow(4), csApsSignalDefectHigh(5), csApsSignalFailureLow(6), csApsSignalFailureHigh(7), csApsForceSwitch(8), csApsLockOut(9)}</td><td>When the working line on one end fails, its other end is told to do an APS switch.
Max-Access: read-only
Status: current
The following switch request types describe, in order of priority, reasons for an APS switchover:
<table>
<tr><th>Switch Request Type</th><th>Description</th></tr>
<tr><td>csApsLockOut</td><td>This request type is the highest priority switch. This will override all other requests.</td></tr>
<tr><td>csApsForce Switch</td><td>This request type is caused by a forced switch that forces hardware to switch the active line even if the other line (the working line or the protection line) is in alarm.</td></tr>
<tr><td>csApsSignal FailureHigh</td><td>This request type occurred because threshold for csApsSigDegradeBER was exceeded.</td></tr>
<tr><td>csApsSignal FailureLow</td><td>The definition is the same as above, but lower priority.</td></tr>
<tr><td>csApsSignal DefectHigh</td><td>This request type occurred because threshold for csApsSigFaultBER was exceeded.</td></tr>
<tr><td>csApsSignal DefectLow</td><td>The definition is the same as above, but lower priority.</td></tr>
<tr><td>csApsManual</td><td>This manual switch causes APS switch unless a request of equal or higher priority is in effect.</td></tr>
<tr><td>csApsRevertive</td><td>This request type switches back to working line after the Wait-to-Restore interval is over, and failures are cleared. It is the lowest priority.</td></tr>
</table></td><td>none</td></tr>
</table>

SONET Section Total Table

The object identifiers are listed in Table 3-39.

Table 3-39 SONET Section Total Table Object Identifiers

Name	Object Identifier
cssTotalTable	::= {csSection 1}
cssTotalEntry	::= {cssTotalTable 1}
cssTotalESs	::= {cssTotalEntry 1}
cssTotalSESs	::= {cssTotalEntry 2}
cssTotalSEFSs	::= {cssTotalEntry 3}
cssTotalCVs	::= {cssTotalEntry 4}

The MIB objects are listed in Table 3-40.

Table 3-40 SONET Section Total Table MIB Objects

Name	Syntax	Description	Default Value
cssTotalTable	`Sequence of cssTotalEntry`	Describes the SONET/SDH Section Total table. It contains the cumulative sum of the various statistics for the 24-hour period preceding the current interval. Since the line was enabled, the object `SONETMediumValidIntervals` from RFC 2558 contains the number of elasped 15-minute intervals. Max-Access: not-accessible Status: current	none
cssTotalEntry	`cssTotalEntry`	Provides an entry in the SONET/SDH Section Total table. Entries are created automatically for SONET lines. Max-Access: not-accessible Status: current The index is `ifIndex`.	none
cssTotalESs	`Gauge32 (in units of errored seconds)`	Detects the number of ES encountered by a SONET/SDH Section in the last 24 hours. Refer to RFC 2558. Max-Access: read-only Status: current	none
cssTotalSESs	`Gauge32 (in units of Severely Errored Seconds)`	Detects the number of SES encountered by a SONET/SDH Section in the last 24 hours. Refer to RFC 2558. Max-Access: read-only Status: current	none

Table 3-40 SONET Section Total Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cssTotalSEFSs	Gauge32 (in units of Severely Errored Framing Seconds)	Detects the number of SEFS encountered by a SONET/SDH Section in the last 24 hours. Refer to RFC 2558. Max-Access: read-only Status: current	none
cssTotalCVs	Gauge32 (in units of Coding Violations)	Detects the number of Coding Violations encountered by a SONET/SDH Section in the last 24 hours. Refer to RFC 2558. Max-Access: read-only Status: current	none

SONET Section Trace Table

The object identifiers are listed in Table 3-41.

Table 3-41 SONET Section Table Object Identifiers

Name	Object Identifier
cssTraceTable	::= {csSection 2}
cssTraceEntry	::= {cssTraceTable 1}
cssTraceToTransmit	::= {cssTraceEntry 1}
cssTraceToExpect	::= {cssTraceEntry 2}
cssTraceFailure	::= {cssTraceEntry 3}
cssTraceReceived	::= {cssTraceEntry 4}

The MIB objects are listed in Table 3-42.

Table 3-42 SONET Section Table MIB Objects

Name	Syntax	Description	Default Value
cssTraceTable	`Sequence of cssTraceEntry`	Describes the SONET/SDH Section Trace table. This table contains objects for tracing the SONET section. Max-Access: not-accessible Status: current	none
cssTraceEntry	`cssTraceEntry`	Provides an entry in the trace table. Entries exist for active SONET lines. The objects in this table are used to verify continued connection between the two ends of the line. Max-Access: not-accessible Status: current The index is `ifIndex`.	none
cssTraceToTransmit	`OCTET STRING (SIZE(0|16|64))`	Describes SONET Section Trace To Transmit string. It is transmitted to perform SONET section trace diagnostics. The trace string is repetitively transmitted so that a trace receiving terminal can verify its continued connection to the intended transmitter. The default value is a zero-length string. Unless this object is set to a nonzero length string, tracing will not be performed. Refer to Bellcore GR-253-CORE. Max-Access: read-write Status: current	none
cssTraceToExpect	`OCTET STRING (SIZE(0|16|64))`	Describes the SONET Section Trace To Expect. The receiving terminal verifies if the incoming string matches this string. The value of `cssTraceFailure` indicates whether a trace mismatch occurred. The default value is a zero-length string. Refer to Bellcore GR-253-CORE. Max-Access: read-write Status: current	none
cssTraceFailure	`TruthValue`	Specifies the value of this object is set to `true` when the SONET section received trace does not match the `cssTraceToExpect` object. Refer to Bellcore GR-253-CORE. Max-Access: read-only Status: current	none
cssTraceReceived	`OCTET STRING (SIZE (0|16|64))`	Views the SONET Section Trace that is received by the receiving terminal. Refer to Bellcore GR-253-CORE. Max-Access: read-only Status: current	none

SONET Line Total Table

The object identifiers are listed in Table 3-43.

Table 3-43 SONET Line Total Table Object Identifiers

Name	Object Identifier
cslTotalTable	::= {csLine 1}
cslTotalEntry	::= {cslTotalTable 1}
cslTotalESs	::= {cslTotalEntry 1}
cslTotalSESs	::= {cslTotalEntry 2}
cslTotalCVs	::= {cslTotalEntry 3}
cslTotalUASs	::= {cslTotalEntry 4}

The MIB objects are listed in Table 3-44.

Table 3-44 SONET Line Total Table MIB Objects

Name	Syntax	Description	Default Value
cslTotalTable	`Sequence of cslTotalEntry`	Describes the 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` contains the number of 15-minute intervals that have elapsed since the line was enabled. Refer to RFC 2558. Max-Access: not-accessible Status: current	none
cslTotalEntry	`cslTotalEntry`	Contains an entry in the SONET/SDH Line Total table. Entries are created automatically for SONET lines. Max-Access: not-accessible Status: current The index is `ifIndex`.	none
cslTotalESs	`Gauge32 (in units of errored seconds)`	Detects the number of ES encountered by a SONET/SDH Line in the last 24 hours. Max-Access: read-only Status: current	none
cslTotalSESs	`Gauge32 (in units of Severely Errored Seconds)`	Detects the number of SES encountered by a SONET/SDH Line in the last 24 hours. Refer to RFC 2558. Max-Access: read-only Status: current	none

Table 3-44 SONET Line Total Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cslTotalCVs	Gauge32 (in units of Coding Violations)	Detects the number of Coding Violations encountered by a SONET/SDH Line in the last 24 hours. Refer to RFC 2558. Max-Access: read-only Status: current	none
cslTotalUASs	Gauge32 (in units of unavailable seconds)	Detects the number of UAS encountered by a SONET/SDH Line in the last 24 hours. Refer to RFC 2558. Max-Access: read-only Status: current	none

SONET Far End Line Total Table

The object identifiers are listed in Table 3-45.

Table 3-45 SONET Far End Line Total Table Object Identifiers

Name	Object Identifier
cslFarEndTotalTable	::= {csLine 2}
cslFarEndTotalEntry	::= {cslFarEndTotalTable 1}
cslFarEndTotalESs	::= {cslFarEndTotalEntry 1}
cslFarEndTotalSESs	::= {cslFarEndTotalEntry 2}
cslFarEndTotalCVs	::= {cslFarEndTotalEntry 3}
cslFarEndTotalUASs	::= {cslFarEndTotalEntry 4}

The MIB objects are listed in Table 3-46.

Table 3-46 SONET Far End Line Total Table MIB Objects

Name	Syntax	Description	Default Value
cslFarEndTotalTable	`Sequence of cslFarEndTotalEntry`	Describes the 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 was enabled. Max-Access: not-accessible Status: current	none
cslFarEndTotalEntry	`cslFarEndTotalEntry`	Provides an entry in the SONET/SDH Far End Line Total table. Entries are created automatically for SONET lines. Max-Access: not-accessible Status: current The index is `ifIndex`.	none
cslFarEndTotalESs	`Gauge32 (in units of errored seconds)`	Detects the number of ES encountered by a SONET/SDH Far End Line in the last 24 hours. Refer to RFC 2558. Max-Access: read-only Status: current	none
cslFarEndTotalSESs	`Gauge32 (in units of Severely Errored Seconds)`	Detects the number of SES encountered by a SONET/SDH Far End Line in the last 24 hours. Refer to RFC 2558. Max-Access: read-only Status: current	none
cslFarEndTotalCVs	`Gauge32 (in units of Coding Violations)`	Detects the number of Coding Violations encountered by a SONET/SDH Far End Line in the last 24 hours. Refer to RFC 2558. Max-Access: read-only Status: current	none
cslFarEndTotalUASs	`Gauge32 (in units of unavailable seconds)`	Detects the number of UAS encountered by a SONET/SDH Far End Line in the last 24 hours. Refer to RFC 2558. Max-Access: read-only Status: current	none

SONET Path Total Table

The object identifiers are listed in Table 3-47.

Table 3-47 SONET Path Total Table Object Identifiers

Name	Object Identifier
cspTotalTable	::= {csPath 1}
cspTotalEntry	::= {cspTotalTable 1}
cspTotalESs	::= {cspTotalEntry 1}
cspTotalSESs	::= {cspTotalEntry 2}
cspTotalCVs	::= {cspTotalEntry 3}
cspTotalUASs	::= {cspTotalEntry 4}

The MIB objects are listed in Table 3-48.

Table 3-48 SONET Path Total Table MIB Objects

Name	Syntax	Description	Default Value
cspTotalTable	`Sequence of cspTotalEntry`	Describes the 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. Max-Access: not-accessible Status: current	none
cspTotalEntry	`cspTotalEntry`	Provides an entry in the SONET/SDH Path Total table. Entries are created automatically for SONET lines. Max-Access: not-accessible Status: current The index is `ifIndex`.	none
cspTotalESs	`Gauge32 (in units of errored seconds)`	Detects the number of ES encountered by a SONET/SDH Path in the last 24 hours. Refer to RFC 2558. Max-Access: read-only Status: current	none
cspTotalSESs	`Gauge32 (in units of Severely Errored Seconds)`	Detects the number of SES encountered by a SONET/SDH Path in the last 24 hours. Refer to RFC 2558. Max-Access: read-only Status: current	none

Table 3-48 SONET Path Total Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cspTotalCVs	Gauge32 (in units of Coding Violations)	Detects the number of Coding Violations encountered by a SONET/SDH Path in the last 24 hours. Refer to RFC 2558. Max-Access: read-only Status: current	none
cspTotalUASs	Gauge32 (in units of unavailable seconds)	Detects the number of UAS encountered by a SONET/SDH Path in the last 24 hours. Refer to RFC 2558. Max-Access: read-only Status: current	none

SONET Far End Path Total Table

The object identifiers are listed in Table 3-49.

Table 3-49 SONET Far End Path Total Table Object Identifiers

Name	Object Identifier
cspFarEndTotalTable	::= {csPath 2}
cspFarEndTotalEntry	::= {cspFarEndTotalTable 1}
cspFarEndTotalESs	::= {cspFarEndTotalEntry 1}
cspFarEndTotalSESs	::= {cspFarEndTotalEntry 2}
cspFarEndTotalCVs	::= {cspFarEndTotalEntry 3}
cspFarEndTotalUASs	::= {cspFarEndTotalEntry 4}

The MIB objects are listed in Table 3-50.

Table 3-50 SONET Far End Path Total Table MIB Objects

Name	Syntax	Description	Default Value
cspFarEndTotalTable	`Sequence of cspFarEndTotalEntry`	Describes the SONET/SDH Far End Path Total table. Far End refers to 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. Max-Access: not-accessible Status: current	none
cspFarEndTotalEntry	`cspFarEndTotalEntry`	Provides an entry in the SONET/SDH Far End Path Total table. Entries are created automatically for SONET lines. Max-Access: not-accessible Status: current The index is `ifIndex`.	none
cspFarEndTotalESs	`Gauge32 (in units of errored seconds)`	Detects the number of Errored Seconds encountered by a SONET/SDH far end path in the last 24 hours. Refer to RFC 2558. Max-Access: read-only Status: current	none
cspFarEndTotalSESs	`Gauge32 (in units of Severely Errored Seconds)`	Detects the number of Severely Errored Seconds encountered by a SONET/SDH far end path in the last 24 hours. Refer to RFC 2558. Max-Access: read-only Status: current	none
cspFarEndTotalCVs	`Gauge32 (in units of Coding Violations)`	Detects the number of Coding Violations encountered by a SONET/SDH far end path in the last 24 hours. Refer to RFC 2558. Max-Access: read-only Status: current	none
cspFarEndTotalUASs	`Gauge32 (in units of unavailable seconds)`	Detects the number of Unavailable Seconds encountered by a SONET/SDH far end path in the last 24 hours. Refer to RFC 2558. Max-Access: read-only Status: current	none

Path Trace Table

The object identifiers are listed in Table 3-51.

Table 3-51 Path Trace Table Object Identifiers

Name	Object Identifier
cspTraceTable	::= {csPath 3}
cspTraceEntry	::= {cspTraceTable 1}
cspTraceToTransmit	::= {cspTraceEntry 1}
cspTraceToExpect	::= {cspTraceEntry 2}
cspTraceFailure	::= { cspTraceEntry 3 }
cspTraceReceived	::= {cspTraceEntry 4}

The MIB objects are listed in Table 3-52.

Table 3-52 Path Trace Table MIB Objects

Name	Syntax	Description	Default Value
cspTraceTable	`Sequence of cspTraceEntry`	Describes the SONET/SDH Path Trace table. This table contains objects for tracing the SONET path. Max-Access: not accessible Status: current	none
cspTraceEntry	`cspTraceEntry`	Provides an entry in the SONET/SDH Path Trace table. The entries exist for active SONET lines. The objects in this table are used to verify continued connection between the two ends of the line. Max-Access: not accessible Status: current The index is `ifIndex`.	none
cspTraceToTransmit	`OCTET STRING (SIZE(0|16|64))`	Describes SONET Path Trace To Transmit. The trace string is repetitively transmitted so that a trace-receiving terminal can verify its continued connection to the intended transmitter. The default value is a zero-length string. Unless this object is set to a nonzero length string, tracing cannot be performed. Refer to Bellcore GR-253-CORE. Max-Access: read-write Status: current	none

Table 3-52 Path Trace Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cspTraceToExpect	OCTET STRING (SIZE(0|16|64))	Describes SONET Path Trace To Expect. The receiving terminal verifies that the incoming string matches this string. The value of cspTraceFailure indicates whether a trace mismatch occurred. The default value is a zero-length string. Refer to Bellcore GR-253-CORE. Max-Access: read-write Status: current	none
cspTraceFailure	TruthValue	Sets the value of this object to true when the trace received by the SONET Path does not match the cspTraceToExpect. Refer to Bellcore GR-253-CORE. Max-Access: read-only Status: current	none
cspTraceReceived	OCTET STRING (SIZE(0|16|64))	Views the SONET Path Trace that is received by the receiving terminal. Refer to Bellcore GR-253-CORE. Max-Access: read-only Status: current	none

SONET Statistics Tables

The object identifiers are listed in Table 3-53.

Table 3-53 SONET Statistics Table Object Identifiers

Name	Object Identifier
csStatsTable	::= {csStats 1}
csStatsEntry	::= {csStatsTable 1}
cssLOSs	::= {csStatsEntry 1}
cssLOFs	::= {csStatsEntry 2}
cslAISs	::= {csStatsEntry 3}
cslRFIs	::= {csStatsEntry 4}
cspAISs	::= {csStatsEntry 5}
cspRFIs	::= {csStatsEntry 6}

The MIB objects are listed in Table 3-54.

Table 3-54 SONET Statistics Table MIB Objects

Name	Syntax	Description	Default Value
csStatsTable	`Sequence of csStatsEntry`	Describes the SONET/SDH Section statistics table. This table maintains the number of times the line encountered LOS, LOF, AIS, and Remote Failure Indications (RFI). Max-Access: not-accessible Status: current	none
csStatsEntry	`csStatsEntry`	Provides an entry in the SONET/SDH statistics table. These are real-time statistics for the SONET section, line, and path layers. The statistics are gathered for each SONET line. An entry is automatically created and is indexed by ifIndex. Max-Access: not-accessible Status: current The index is `ifIndex`.	none
cssLOSs	`Counter32 (in units of loss of signals)`	Detects the number of LOS encountered by a SONET/SDH Section. A high value for this object can indicate a problem with the SONET Section layer. Refer to RFC 2558. Max-Access: read-only Status: current	none
cssLOFs	`Counter32 (in units of loss of frames)`	Detects the number of LOF encountered by a SONET/SDH Section. A high value for this object may indicate a problem with the SONET Section layer. Refer to RFC 2558. Max-Access: read-only Status: current	none
cslAISs	`Counter32 (in units of Alarm Indication Signals)`	Detects the number of AIS encountered by a SONET/SDH Line. A high value for this object may indicate a problem with the SONET Line layer. Refer to RFC 2558. Max-Access: read-only Status: current	none
cslRFIs	`Counter32 (in units of Remote Failure Indications)`	Detects the number of RFI encountered by a SONET/SDH Line. A high value for this object may indicate a problem with the SONET Line layer. Refer to RFC 2558. Max-Access: read-only Status: current	none

Table 3-54 SONET Statistics Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cspAISs	Counter32 (in units of Alarm Indication Signals)	Detects the number of AIS encountered by a SONET/SDH Path. A high value for this object may indicate a problem with the SONET Path layer. Refer to RFC 2558. Max-Access: read-only Status: current	none
cspRFIs	Counter32 (in units of Remote Failure Indications)	Detects the number of RFI encountered by a SONET/SDH Path. A high value for this object may indicate a problem with the SONET Path layer. Refer to RFC 2558. Max-Access: read-only Status: current	none

Cisco SONET Conformance Information

One object identifier is listed in Table 3-55.

Table 3-55 Cisco SONET Conformance Information Object Identifier

Name	Object Identifier
ciscoSONETMIBCompliance	::= {ciscoSONETMIBCompliances 1}

One MIB object is listed in Table 3-56.

Table 3-56 Cisco SONET Conformance Information MIB Object

Name	Mandatory Groups	Description	Default Value
ciscoSONETMIB Compliance	ciscoSONETConfMIBGroup, ciscoSONETStatsMIBGroup, ciscoSONETTraceMIBGroup	Specifies the compliance statement for the ciscoSONET management group. It is required if SONET APS is supported. Status: current Module: AXSM Group: ciscoSONETApsMIBGroup	none

Cisco SONET Units of Conformance

The object identifiers are listed in Table 3-57.

Table 3-57 Cisco SONET Units of Conformance Object Identifiers

Name	Object Identifier
ciscoSONETConfMIBGroup	::= {ciscoSONETMIBGroups 1}
ciscoSONETStatsMIBGroup	::= {ciscoSONETMIBGroups 2}
ciscoSONETTraceMIBGroup	::= {ciscoSONETMIBGroups 3}
c**iscoSONETApsMIBGroup**	::= {ciscoSONETMIBGroups 4}

The MIB objects are listed in Table 3-58.

Table 3-58 Cisco SONET Units of Conformance MIB Objects

Name	Objects	Description	Default Value
ciscoSONETConfMIB Group	csConfigLoopbackType, csConfigXmtClockSource, csConfigFrameScramble, csConfigType	Specifies the objects used for configuring SONET lines. Status: current	none
ciscoSONETStatsMIB Group	cssTotalESs, cssTotalSESs, cssTotalSEFSs, cssTotalCVs, cssLOSs, cssLOFs, cslAISs, cslRFIs, cspAISs, cspRFIs, cslTotalESs, cslTotalSESs, cslTotalCVs, cslTotalUASs, cslFarEndTotalESs, cslFarEndTotalSESs, cslFarEndTotalCVs, cslFarEndTotalUASs, cspTotalESs, cspTotalSESs, cspTotalCVs, cspTotalUASs, cspFarEndTotalESs, cspFarEndTotalSESs, cspFarEndTotalCVs, cspFarEndTotalUASs	Relates to the SONET line statistics. Status: current	none
ciscoSONETTrace MIBGroup	cssTraceToTransmit, cssTraceToExpect, cssTraceFailure, cspTraceToTransmit, cspTraceToExpect, cspTraceFailure	Relates to the SONET line connectivity. Status: current	none
c**iscoSONETApsMIB Group**	csApsProtectionIndex, csApsEnable, csApsArchMode, csApsActiveLine, csApsSigFaultBER, csApsSigDegradeBER, csApsWaitToRestore, csApsDirection, csApsRevertive, csApsLineFailureCode, csApsLineSwitchReason	Configures APS over SONET lines. Status: current	none

Cisco WAN SONET

This section describes the Cisco WAN SONET MIB objects that are defined in the file CISCO-WAN-SONET-MIB.my. This MIB file describes the SONET/SDH interface objects. The objects are an extension to the standard SONET MIB, as described in RFC 2558.

Note The Cisco WAN SONET MIB is supported by AXSM and defined under the StrataCom Enterprise.

The MIB objects used for Cisco WAN SONET include:

- SONET Section Alarm Table
- Line Alarm Table
- Path Alarm Table
- Cisco WAN SONET Units of Conformance

SONET Section Alarm Table

The SONET/SDH Section Alarm table contains objects for setting alarm thresholds and reading alarm severity.

The object identifiers are listed in Table 3-59.

Table 3-59 SONET Section Alarm Table Object Identifiers

Name	Object Identifier
cwsSectionAlarmTable	::= {cwsSection 1}
cwsSectionAlarmEntry	::= {cwsSectionAlarmTable 1}
cwsSectionStatisticalAlarmSeverity	::= {cwsSectionAlarmEntry 1}
cwsSectionCurrentESsThreshold	::= {cwsSectionAlarmEntry 2}
cwsSectionTotalESsThreshold	::= {cwsSectionAlarmEntry 3}
cwsSectionCurrentSESsThreshold	::= {cwsSectionAlarmEntry 4}
cwsSectionTotalSESsThreshold	::= {cwsSectionAlarmEntry 5}
cwsSectionCurrentSEFSsThreshold	::= {cwsSectionAlarmEntry 6}
cwsSectionTotalSEFSsThreshold	::= {cwsSectionAlarmEntry 7}
cwsSectionCurrentCVsThreshold	::= {cwsSectionAlarmEntry 8}
cwsSectionTotalCVsThreshold	::= {cwsSectionAlarmEntry 9}
cwsSectionStatAlarmStatus	::= {cwsSectionAlarmEntry 10}

The MIB objects are listed in Table 3-60.

Table 3-60 SONET Section Alarm Table MIB Objects

Name	Syntax	Description	Default Value
cwsSectionAlarm Table	`Sequence of cwsSectionAlarmEntry`	Contains the objects for setting alarm thresholds and reading alarm severity. Max-Access: not-accessible Status: current	none
cwsSectionAlarm Entry	`cwsSectionAlarmEntry`	Provides an entry in the alarm table. The entries exist for active SONET lines, with default threshold values. Thresholds may be configured for ES, SES, SEFS, and CV for section layer SONET lines. Once these thresholds are exceeded, a minor or major alarm is raised. Refer to RFC 2558 for descriptions of ES, SES, SEFS, and CV. Max-Access: not-accessible Status: current The index is `ifIndex`.	none
cwsSectionStatistical AlarmSeverity	`INTEGER {minor(1), major(2)}`	Indicates whether exceeding this statistical threshold is a minor or major severity alarm. Max-Access: read-write Status: current	`minor`
cwsSectionCurrent ESsThreshold	`Unsigned32 (0..2147483647)`	Indicates the threshold for ES, for the current 15-minute interval, which an alarm is generated. Refer to RFC 2558. Max-Access: read-write Status: current	`20`
cwsSectionTotalESs Threshold	`Unsigned32 (0..2147483647)`	Indicates the ES threshold, for the 24-hour interval, which an alarm is generated. Refer to RFC 2558. Max-Access: read-write Status: current	`20`
cwsSectionCurrent SESsThreshold	`Unsigned32 (0..2147483647) (in units of Severely Errored Seconds)`	Indicates the threshold for SES for the current 15-minute interval, which an alarm is generated. Refer to RFC 2558. Max-Access: read-write Status: current	`3`

Table 3-60 SONET Section Alarm Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwsSectionTotal SESsThreshold	Unsigned32 (0..2147483647) (in units of Severely Errored Seconds)	Indicates the threshold for SES for the 24-hour interval. If this threshold is exceeded, an alarm is generated. Refer to RFC 2558. Max-Access: read-write Status: current	3
cwsSectionCurrent SEFSsThreshold	Unsigned32 (0..2147483647) (in units of Severely Errored Framing Seconds)	Indicates the SEFS threshold. If this threshold is exceeded, an alarm is generated. Refer to RFC 2558. Max-Access: read-write Status: current	3
cwsSectionTotal SEFSsThreshold	Unsigned32 (0..2147483647) (in units of Severely Errored Framing Seconds)	Indicates the SEFS threshold for the 24-hour interval. If this threshold is exceeded, an alarm is generated. Refer to RFC 2558. Max-Access: read-write Status: current	3
cwsSectionCurrent CVsThreshold	Unsigned32 (0..2147483647) (in units of number of Coding Violations)	Indicates the threshold for CV. If this threshold is exceeded, an alarm is generated. Refer to RFC 2558. Max-Access: read-write Status: current	25
cwsSectionTotalCVs Threshold	Unsigned32 (0..2147483647) (in units of number of Coding Violations)	Indicates the threshold for CV for a 24-hour interval. If this threshold is exceeded, an alarm is generated. Refer to RFC 2558. Max-Access: read-write Status: current	25

Table 3-60 SONET Section Alarm Table MIB Objects (continued)

<table>
<tr><th>Name</th><th>Syntax</th><th>Description</th><th>Default Value</th></tr>
<tr><td>cwsSectionStatAlarm Status</td><td>Unsigned32 (1..511)</td><td>Indicates the status of the interface. When an alarm is generated, this object would indicate which threshold was exceeded. This object is updated every 15 minutes. Once the current 15-minute window has passed, and there are no errors in the current 15-minute window, the object is cleared from the lower level SONET task. The cwsSectionStatAlarmStatus is a bit map represented as a sum, therefore, it can represent multiple defects simultaneously. If none of the bits are set, it indicates no defect.
Max-Access: read-write
Status: current
The following are the defects indicated by the various bit positions:
<table>
<tr><th>Bit Position</th><th>Defect</th></tr>
<tr><td>1</td><td>SONETSectionTotalESs</td></tr>
<tr><td>2</td><td>SONETSectionTotalSESs</td></tr>
<tr><td>3</td><td>SONETSectionTotalSEFSs</td></tr>
<tr><td>4</td><td>SONETSectionTotalCVs</td></tr>
<tr><td>5</td><td>SONETSectionCurrentESs</td></tr>
<tr><td>6</td><td>SONETSectionCurrentSESs</td></tr>
<tr><td>7</td><td>SONETSectionCurrentSEFSs</td></tr>
<tr><td>8</td><td>SONETSectionCurrentCVs</td></tr>
</table></td><td>none</td></tr>
</table>

Line Alarm Table

The object identifiers are listed in Table 3-61.

Note Changing any threshold effects both the Near End and the Far End threshold.

Table 3-61 Line Alarm Table Object Identifiers

Name	Object Identifier
cwsLineAlarmTable	::= {cwsLine 1}
cwsLineAlarmEntry	::= {cwsLineAlarmTable 1}
cwsLineStatisticalAlarmSeverity	::= {cwsLineAlarmEntry 1}
cwsLineCurrentESsThreshold	::= {cwsLineAlarmEntry 2}
cwsLineTotalESsThreshold	::= {cwsLineAlarmEntry 3}
cwsLineCurrentSESsThreshold	::= {cwsLineAlarmEntry 4}
cwsLineTotalSESsThreshold	::= {cwsLineAlarmEntry 5}
cwsLineCurrentCVsThresholdE	::= {cwsLineAlarmEntry 6}
cwsLineTotalCVsThreshold	::= {cwsLineAlarmEntry 7}
cwsLineCurrentUASsThreshold	::= {cwsLineAlarmEntry 8}
cwsLineTotalUASsThreshold	::= {cwsLineAlarmEntry 9}
cwsLineStatAlarmStatus	::= {cwsLineAlarmEntry 10}

The MIB objects are listed in Table 3-62.

Table 3-62 Line Alarm Table MIB Objects

Name	Syntax	Description	Default Value
cwsLineAlarmTable	`Sequence of cwsLineAlarmEntry`	Describes the SONET/SDH Line Alarm table. This table contains objects which are thresholds that the user can set. Once a threshold is exceeded, an alarm is flagged. Changing any threshold will affect both the Near End and Far End (remote end of the line) threshold. Max-Access: not-accessible Status: current	none
cwsLineAlarmEntry	`cwsLineAlarmEntry`	Provides an entry in the SONET/SDH Line Alarm table. The entries may be created for an `ifType` value of `SONET(39)`. Thresholds can be specified for ES, SES, SEFS, CV, and UAS for line layer of SONET lines. Once these thresholds are exceeded, a minor or major alarm is raised. Refer to RFC 2558 for descriptions of ES, SES, CV, and UAS. Max-Access: not-accessible Status: current The index is `ifIndex`.	none

Table 3-62 Line Alarm Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwsLineStatistical AlarmSeverity	INTEGER {minor(1), major(2)}	If any statistical threshold is exceeded, this object is set to major or minor. Max-Access: read-write Status: current	none
cwsLineCurrentESs Threshold	Unsigned32 (0..2147483647)	Indicates the threshold for ES for the current 15-minute interval. If this threshold is exceeded, an alarm is generated. This threshold is the same for Far End and Near End. Refer to RFC 2558. Max-Access: read-write Status: current	20
cwsLineTotalESs Threshold	Unsigned32 (0..2147483647)	Indicates the ES threshold for the 24-hour interval. If this threshold is exceeded, an alarm is generated. This threshold is the same for Far End and Near End. Refer to RFC 2558. Max-Access: read-write Status: current	20
cwsLineCurrent SESsThreshold	Unsigned32 (0..2147483647)	Specifies the SES threshold for the current 15-minute interval. If this threshold is exceeded, an alarm is generated. This threshold is the same for Far End and Near End. Refer to RFC 2558. Max-Access: read-write Status: current	3
cwsLineTotalSESs Threshold	Unsigned32 (0..2147483647)	Specifies the SES threshold for the 24-hour interval. If this threshold is exceeded, an alarm is generated. This threshold is the same for Far End and Near End. Refer to o RFC 2558. Max-Access: read-write Status: current	3
cwsLineCurrent CVsThresholdE	Unsigned32 (0..2147483647)	Specifies the CV threshold for the current 15-minute interval. If this threshold is exceeded, an alarm is generated. This threshold is the same for Far End and Near End. Refer to RFC 2558. Max-Access: read-write Status: current	3

Table 3-62 Line Alarm Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwsLineTotalCVs Threshold	Unsigned32 (0..2147483647)	Specifies the CV threshold for the 24-hour interval. If this threshold is exceeded, an alarm is generated. This threshold is the same for Far End and Near End. Refer to RFC 2558. Max-Access: read-write Status: current	3
cwsLineCurrent UASsThreshold	Unsigned32 (0..2147483647)	Specifies the UAS threshold for the current 15-minute interval. If this threshold is exceeded, an alarm is generated. This threshold is the same for Far End and Near End. Refer to RFC 2558. Max-Access: read-write Status: current	none
cwsLineTotalUASs Threshold	Unsigned32 (0..2147483647)	Specifies the UAS threshold for the 24-hour interval. If this threshold is exceeded, an alarm is generated. This threshold is the same for Far End and Near End. Refer to RFC 2558. Max-Access: read-write Status: current	none

Table 3-62 Line Alarm Table MIB Objects (continued)

Name	Syntax	Description		Default Value
cwsLineStatAlarm Status	`Unsigned32`	Indicates the status of the SONET line. This object is updated every 15 minutes. Once the current 15-minute window has passed, and if there are no errors in the current 15-minute window, the object is cleared from the lower level SONET task. The `cwsLineStatAlarmStatus` is a bit map represented as a sum, so it can represent multiple defects simultaneously. If none of the bits are set, it indicates no defect. Max-Access: read-only Status: current The defects indicated by the bit position include:		none
		Bit Position	**Defect**	
		1	`SONETLineTotalESs`	
		2	`SONETLineTotalSESs`	
		3	`SONETLineTotalCVs`	
		4	`SONETLineTotalUASs`	
		5	`SONETLineCurrentESs`	
		6	`SONETLineCurrentSESs`	
		7	`SONETLineCurrentCVs`	
		8	`SONETLineCurrentUASs`	
		9	`SONETFarEndLineCurrentESs`	
		10	`SONETFarEndLineTotalESs`	
		11	`SONETFarEndLineTotalSESs`	
		12	`SONETFarEndLineTotalCVs`	
		13	`SONETFarEndLineTotalUASs`	
		14	`SONETFarEndLineCurrentSESs`	
		15	`SONETFarEndLineCurrentCVs`	
		16	`SONETFarEndLineCurrentUASs`	

Path Alarm Table

The object identifiers are listed in Table 3-63.

Table 3-63 Path Alarm Table Object Identifiers

Name	Object Identifier
cwsPathAlarmTable	::= {cwsPath 1}
cwsPathAlarmEntry	::= {cwsPathAlarmTable 1}
cwsPathStatisticalAlarmSeverity	::= {cwsPathAlarmEntry 1}
cwsPathCurrentESsThreshold	::= {cwsPathAlarmEntry 2}
cwsPathTotalESsThreshold	::= {cwsPathAlarmEntry 3}
cwsPathCurrentSESsThreshold	::= {cwsPathAlarmEntry 4}
cwsPathTotalSESsThreshold	::= {cwsPathAlarmEntry 5}
cwsPathCurrentCVsThreshold	::= {cwsPathAlarmEntry 6}
cwsPathTotalCVsThreshold	::= {cwsPathAlarmEntry 7}
cwsPathCurrentUASsThreshold	::= {cwsPathAlarmEntry 8}
cwsPathTotalUASsThreshold	::= {cwsPathAlarmEntry 9}
cwsPathStatAlarmStatus	::= {cwsPathAlarmEntry 10}

The MIB objects are listed in Table 3-64.

Table 3-64 Path Alarm Table MIB Objects

Name	Syntax	Description	Default Value
cwsPathAlarmTable	`Sequence of cwsPathAlarmEntry`	Describes the SONET/SDH Path Alarm table. This table contains objects for thresholds which a user can set. When any threshold is exceeded, an alarm is set. Changing any threshold effects both the Near End and Far End threshold. Max-Access: not-accessible Status: current	none
cwsPathAlarmEntry	`cwsPathAlarmEntry`	Provides an entry in the SONET/SDH Path Alarm table. Thresholds can be specified for ES, SES, SEFS, CV, and UAS for path layer of SONET lines. Once these thresholds are exceeded, a minor or major alarm is raised. Refer to RFC 2558 for descriptions of ES, SES, SEFS, CV, and UAS. Max-Access: not-accessible Status: current The index is `ifIndex`.	none

Table 3-64 Path Alarm Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwsPathStatistical AlarmSeverity	`INTEGER {minor(1), major(2)}`	When any statistical threshold is exceeded, this object is set to `major` or `minor`. Max-Access: read-write Status: current	none
cwsPathCurrentESs Threshold	`Unsigned32 (0..2147483647)`	Indicates the threshold for ES for the current 15-minute interval. If this threshold is exceeded, an alarm is generated. This threshold is the same for Far End and Near End. Refer to RFC 2558. Max-Access: read-write Status: current	`20`
cwsPathTotalESs Threshold	`Unsigned32 (0..2147483647)`	Indicates the threshold for ES for the 24-hour interval. If this threshold is exceeded, an alarm is generated. This threshold is the same for Far End and Near End. Refer to RFC 2558. Max-Access: read-write Status: current	`20`
cwsPathCurrentSESs Threshold	`Unsigned32 (0..2147483647)`	Indicates the threshold for SES for the current 15-minute interval. If this threshold is exceeded, an alarm is generated. This threshold is the same for Far End and Near End. Refer to RFC 2558. Max-Access: read-write Status: current	`3`
cwsPathTotalSESs Threshold	`Unsigned32 (0..2147483647)`	Indicates the threshold for SES for the 24-hour interval. If this threshold is exceeded, an alarm is generated. This threshold is the same for Far End and Near End. Refer to RFC 2558. Max-Access: read-write Status: current	`3`
cwsPathCurrentCVs Threshold	`Unsigned32 (0..2147483647)`	Specifies the Coding Violations threshold for the current 15-minute interval. If this threshold is exceeded, an alarm is generated. This threshold is the same for Far End and Near End. Refer to RFC 2558. Max-Access: read-write Status: current	`25`

Table 3-64 Path Alarm Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwsPathTotalCVs Threshold	Unsigned32 (0..2147483647)	Specifies the Coding Violations threshold for the 24-hour interval. If this threshold is exceeded, an alarm is generated. This threshold is the same for Far End and Near End. Refer to RFC 2558. Max-Access: read-write Status: current	25
cwsPathCurrentUASs Threshold	Unsigned32 (0..2147483647)	Specifies the UAS threshold for the current 15-minute interval. If this threshold is exceeded, an alarm is generated. This threshold is the same for Far End and Near End. Refer to RFC 2558. Max-Access: read-write Status: current	10
cwsPathTotalUASs Threshold	Unsigned32 (0..2147483647)	Specifies the UAS threshold for the 24-hour interval. If this threshold is exceeded, an alarm is generated. This threshold is the same for Far End and Near End. Refer to RFC 2558. Max-Access: read-write Status: current	10

Table 3-64 Path Alarm Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwsPathStatAlarmStatus	Unsigned32	Indicates the status of the interface. The cwsPathStatAlarmStatus is a bit map represented as a sum, so it can represent multiple defects simultaneously. If none of the bits are set, it indicates no defect. Max-Access: read-only Status: current The defects indicated by the bit positions include:	none

Bit Position	Defect
1	SONETPathTotalESs
2	SONETPathTotalSESs
3	SONETPathTotalCVs
4	SONETPathTotalUASs
5	SONETPathCurrentESs
6	SONETPathCurrentSESs
7	SONETPathCurrentCVs
8	SONETPathCurrentUASs
9	SONETFarEndPathTotalESs
10	SONETFarEndPathTotalSESs
11	SONETFarEndPathTotalCVs
12	SONETFarEndPathTotalUASs
13	SONETFarEndPathCurrentESs
14	SONETFarEndPathCurrentSESs
15	SONETFarEndPathCurrentCVs
16	SONETFarEndPathCurrentUASs

Cisco WAN SONET Units of Conformance

One object is listed in Table 3-65.

Table 3-65 Cisco WAN SONET Units of Conformance Object Identifier

Name	Object Identifier
ciscoWANSONETAlarmMIBGroup	::= {ciscoWANSONETMIBGroups 1}

One MIB object is listed in Table 3-66.

Table 3-66 Cisco WAN SONET Section Alarm Table MIB Objects

Name	Objects	Description	Default Value
ciscoWANSONETAlarmMIB Group	cwsSectionStatisticalAlarmSeverity, cwsSectionCurrentESsThreshold, cwsSectionTotalESsThreshold, cwsSectionCurrentSESsThreshold, cwsSectionTotalSESsThreshold, cwsSectionCurrentSEFSsThreshold, cwsSectionTotalSEFSsThreshold, cwsSectionCurrentCVsThreshold, cwsSectionTotalCVsThreshold, cwsSectionStatAlarmStatus, cwsLineStatisticalAlarmSeverity, cwsLineCurrentESsThreshold, cwsLineTotalESsThreshold, cwsLineCurrentSESsThreshold, cwsLineTotalSESsThreshold, cwsLineCurrentCVsThreshold, cwsLineTotalCVsThreshold, cwsLineCurrentUASsThreshold, cwsLineTotalUASsThreshold, cwsLineStatAlarmStatus, cwsPathStatisticalAlarmSeverity, cwsPathCurrentESsThreshold, cwsPathTotalESsThreshold, cwsPathCurrentSESsThreshold, cwsPathTotalSESsThreshold, cwsPathCurrentCVsThreshold, cwsPathTotalCVsThreshold, cwsPathCurrentUASsThreshold, cwsPathTotalUASsThreshold, cwsPathStatAlarmStatus	Relates to the ciscoSONET alarm group. Status: current	none

Cisco WAN Resource Partition

This section describes the MIB module that manages resource partition objects. A resource partition is configured on a virtual interface, which in turn, is configured on a physical line. Through these MIB tables, an administrator can partition connection-related resources like VPI/VCI ranges, bandwidth, and total amount of available connection entries in the switch. The Cisco WAN Resource Partition MIB resides in the CISCO-WAN-RSRC-PART-MIB.my file.

For information about Virtual Switch Interface (VSI), VSI Master, and Controller terms, see Table 4-49.

Note The Cisco WAN Resource Partition MIB is supported by AXSM and RPM, which is also implemented in PXM45.

The Cisco WAN resource partition MIB objects include:

- Resource Partition MIB
- cwRsrcPartIlmi Table
- cwRsrcPartCtlrConf Table
- Cisco WAN Resource Partition Conformance Information

Resource Partition MIB

The object identifiers are listed in Table 3-67.

Table 3-67 Resource Partition MIB Object Identifiers

Name	Object Identifier
cwRsrcPartConfTable	::= {cwRsrcPartConfGrp 1}
cwRsrcPartConfEntry	::= {cwRsrcPartConfTable 1}
cwRsrcPartID	::= {cwRsrcPartConfEntry 1}
cwRsrcPartController	::= {cwRsrcPartConfEntry 2}
cwRsrcPartEgrGuarPctBwConf	::= {cwRsrcPartConfEntry 3}
cwRsrcPartEgrMaxPctBwConf	::= {cwRsrcPartConfEntry 4}
cwRsrcPartIngGuarPctBwConf	::= {cwRsrcPartConfEntry 5}
cwRsrcPartIngMaxPctBwConf	::= {cwRsrcPartConfEntry 6}
cwRsrcPartEgrPctBwUsed	::= {cwRsrcPartConfEntry 7}
cwRsrcPartIngPctBwUsed	::= {cwRsrcPartConfEntry 8}
cwRsrcPartEgrPctBwAvail	::= {cwRsrcPartConfEntry 9}
cwRsrcPartIngPctBwAvail	::= {cwRsrcPartConfEntry 10}
cwRsrcPartVpiLo	::= {cwRsrcPartConfEntry 11}
cwRsrcPartVpiHigh	::= {cwRsrcPartConfEntry 12}
cwRsrcPartVciLo	::= {cwRsrcPartConfEntry 13}
cwRsrcPartVciHigh	::= {cwRsrcPartConfEntry 14}

Table 3-67 Resource Partition MIB Object Identifiers (continued)

Name	Object Identifier
cwRsrcPartGuarCon	::= {cwRsrcPartConfEntry 15}
cwRsrcPartMaxCon	::= {cwRsrcPartConfEntry 16}
cwRsrcPartUsedCon	::= {cwRsrcPartConfEntry 17}
cwRsrcPartAvailCon	::= {cwRsrcPartConfEntry 18}
cwRsrcPartRowStatus	::= {cwRsrcPartConfEntry 19}

The MIB objects are listed in Table 3-68.

Table 3-68 Resource Partition MIB Objects

Name	Syntax	Description	Default Value
cwRsrcPartConfTable	`SEQUENCE OF CwRsrcPartConfEntry`	Configures resource partition parameters. A resource partition is configured on a interface whose `ifType` is `ATMVirtual (149)` or `ATM (37)`. An entry can be added, deleted, or modified. Entries in this table can also be modified through the commands provided by the Command Line Interface (CLI). Max-Access: not-accessible Status: current	none
cwRsrcPartConfEntry	`CwRsrcPartConfEntry`	Provides an entry for a resource partition. A resource partition must be created before connections can be added to the interface. Max-Access: not-accessible Status: current The indexes include: • `ifIndex` • `cwRsrcPartID` • `cwRsrcPartController`	none
cwRsrcPartID	`Unsigned32 (1.. 255)`	Specifies the resource partition identifier. One interface contains multiple resource partitions. Max-Access: not-accessible Status: current	none
cwRsrcPartController	`Unsigned32 (1.. 255)`	Specifies the controller identifier. Once a resource partition is added, the identified controller can access resources configured for this partition. The user can configure redundant controllers. The valid values for controller IDs are from 1 to 5. Max-Access: not-accessible Status: current	none

Table 3-68 Resource Partition MIB Objects (continued)

Name	Syntax	Description	Default Value
cwRsrcPartEgrGuarPctBwConf	Unsigned32 (0..1000000)	Specifies the guaranteed percentage bandwidth reserved for the resource partition in egress direction. The sum of all resource partition guaranteed bandwidths cannot exceed 100% of the interface guaranteed bandwidth. Units: 0.0001 percentage Max-Access: read-create Status: current	0
cwRsrcPartEgrMaxPctBwConf	Unsigned32 (0..1000000)	Specifies the maximum percentage bandwidth for the resource partition in egress direction. This is not the guaranteed bandwidth. If bandwidth is available, the resource partition allocates bandwidth up to this value. Units: 0.0001 percentage Max-Access: read-create Status: current	0
cwRsrcPartIngGuarPctBwConf	Unsigned32 (0..1000000)	Specifies the guaranteed percentage bandwidth reserved for the resource partition in ingress direction. The sum of all resource partition guaranteed bandwidths cannot exceed 100% of the interface guaranteed bandwidth. Units: 0.0001 percentage Max-Access: read-create Status: current	0
cwRsrcPartIngMaxPctBwConf	Unsigned32 (0..1000000)	Specifies the maximum percentage bandwidth allocated for the resource partition in ingress direction. This is not the guaranteed bandwidth. If bandwidth is available, the resource partition allocates bandwidth up to this value. Units: 0.0001 percentage Max-Access: read-create Status: current	0
cwRsrcPartEgrPctBwUsed	Unsigned32 (0..1000000)	Specifies the percentage bandwidth used by the resource partition in egress direction. Units: 0.0001 percentage Max-Access: read-only Status: current	none
cwRsrcPartIngPctBwUsed	Unsigned32 (0..1000000)	Specifies the percentage bandwidth used by the resource partition in ingress direction. Units: 0.0001 percentage Max-Access: read-only Status: current	none

Table 3-68 Resource Partition MIB Objects (continued)

Name	Syntax	Description	Default Value
cwRsrcPartEgrPctBwAvail	`Unsigned32 (0..1000000)`	Specifies the percentage bandwidth available on the resource partition in the egress direction. Units: 0.0001 percentage Max-Access: read-only Status: current	none
cwRsrcPartIngPctBwAvail	`Unsigned32 (0..1000000)`	Specifies the percentage bandwidth available on the resource partition in the ingress direction. Units: 0.0001 percentage Max-Access: read-only Status: current	none
cwRsrcPartVpiLo	`Unsigned32 (0..4095)`	Determines the beginning of the Virtual Path Identifier (VPI) range for this partition. Connections provisioned on this partition must have a VPI greater than or equal to this value. Max-Access: read-create Status: current	none
cwRsrcPartVpiHigh	`Unsigned32 (0..4095)`	Determines the end of the VPI range for this partition. Connections provisioned on this partition must have a VPI less than or equal to this value. Max-Access: read-create Status: current	none
cwRsrcPartVciLo	`Unsigned32 (0..65535)`	Determines the beginning of the Virtual Channel Identifier (VCI) range for this partition. Connections provisioned on this partition must have VCI greater than or equal to this VCI. Max-Access: read-create Status: current	none
cwRsrcPartVciHigh	`Unsigned32 (0..65535)`	Determines the end of the VCI range reserved for this partition. Connections provisioned on this partition must have a VCI less than or equal to this VCI. Max-Access: read-create Status: current	none
cwRsrcPartGuarCon	`Unsigned32 (0..131072)`	Determines the guaranteed number of connections that is configured on this partition. Max-Access: read-create Status: current	`0`

Table 3-68 Resource Partition MIB Objects (continued)

Name	Syntax	Description	Default Value
cwRsrcPartMaxCon	`Unsigned32 (0..131072)`	Determines the maximum number of connections that is configured on this partition. Max-Access: read-create Status: current	none
cwRsrcPartUsedCon	`Unsigned32 (0..131072)`	Determines the number of connections currently in use on this partition. Max-Access: read-only Status: current	none
cwRsrcPartAvailCon	`Unsigned32 (0..131072)`	Determines the number of connections available that can be added on this partition. Max-Access: read-only Status: current	none
cwRsrcPartRowStatus	`RowStatus`	Adds, deletes, or modifies the partition. A resource partition is created by setting this object to `createAndGo(4)`. A row is deleted by setting this object to `destroy(6)`. Max-Access: read-create Status: current	none

cwRsrcPartIlmi Table

The object identifiers are listed in Table 3-69.

Table 3-69 cwRsrcPartIlmi Table Object Identifiers

Name	Object Identifier
cwRsrcPartIlmiTable	::= {cwRsrcPartConfGrp 2}
cwRsrcPartIlmiEntry	::= {cwRsrcPartIlmiTable 1}
cwRsrcPartIlmiEnabled	::= {cwRsrcPartIlmiEntry 1}
cwRsrcPartSignallingVpi	::= {cwRsrcPartIlmiEntry 2}
cwRsrcPartSignallingVci	::= {cwRsrcPartIlmiEntry 3}
cwRsrcPartIlmiTrapEnable	::= {cwRsrcPartIlmiEntry 4}
cwRsrcPartIlmiEstablishConPollIntvl	:= {cwRsrcPartIlmiEntry 5}
cwRsrcPartIlmiCheckConPollIntvl	::= {cwRsrcPartIlmiEntry 6}
cwRsrcPartIlmiConPollInactFactor	::= { cwRsrcPartIlmiEntry 7 }

The MIB objects are listed in Table 3-70.

Table 3-70 cwRsrcPartIlmi Table MIB Objects

Name	Syntax	Description	Default Value
cwRsrcPartIlmiTable	`Sequence of CwRsrcPartIlmiEntry`	Specifies the Interim Local Management Interface (ILMI) configuration table. When a row is created in `cwRsrcPartConfTable`, a corresponding row is automatically created in this table, with the default values for ILMI objects (for example, with ILMI signaling disabled). Max-Access: not-accessible Status: current	none
cwRsrcPartIlmiEntry	`CwRsrcPartIlmiEntry`	Provides an entry in this table that is automatically created when a resource partition is created. By default, ILMI is disabled on active resource partitions. ILMI can be enabled by setting `cwRsrcPartIlmiEnabled` to true. An entry will be deleted from this table when a corresponding resource partition entry is deleted from `cwRsrcPartConfTable`. Max-Access: not-accessible Status: current The indexes include: • `ifIndex` • `cwRsrcPartID`	none
cwRsrcPartIlmiEnabled	`TruthValue`	By setting the value of this object to true, ILMI signaling is enabled on the resource partition. Max-Access: read-write Status: current	`false`
cwRsrcPartSignallingVpi	`INTEGER (0..255)`	Indicates the VPI on which signaling cells arrive on this resource partition. Max-Access: read-write Status: current	`0`
cwRsrcPartSignallingVci	`INTEGER (0..65535)`	Indicates the VCI on which signaling cells arrive on this resource partition. Max-Access: read-write Status: current	`16`
cwRsrcPartIlmiTrap Enable	`TruthValue`	Enables or disables the ILMI trap generation using this object. Max-Access: read-write Status: current	none

Table 3-70 cwRsrcPartIlmi Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwRsrcPartIlmiEstablishConPollIntvl	`Unsigned32 (1..65535) (in units of seconds)`	Specifies the amount of time between successive transmissions of ILMI messages on this interface for the purpose of detecting establishment of ILMI connectivity. Refer to *ATM Forum ILMI 4.0*, Section 8.3.1. Max-Access: read-write Status: current	`1`
cwRsrcPartIlmiCheckConPollIntvl	`Unsigned32 (0..65535) (in units of seconds)`	Specifies the amount of time between successive transmissions of ILMI messages on this interface for the purpose of detecting loss of ILMI connectivity. The value 0 disables ILMI connectivity procedures on this interface. Refer to *ATM Forum ILMI 4.0*, Section 8.3.1. Max-Access: read-write Status: current	`5`
cwRsrcPartIlmiConPollInactFactor	`Unsigned32 (0..65535)`	Determines the number of consecutive polls on this interface for which no ILMI response message is received before ILMI connectivity is declared lost. Refer to *ATM Forum ILMI 4.0*, Section 8.3.1. Max-Access: read-write Status: current	`4`

cwRsrcPartCtlrConf Table

The object identifiers are listed in Table 3-71.

Table 3-71 cwRsrcPartCtlrConf Table Object Identifiers

Name	Object Identifier
cwRsrcPartCtlrConfTable	::= {cwRsrcPartConfGrp 3}
cwRsrcPartCtlrConfEntry	::= {cwRsrcPartCtlrConfTable 1}
cwRsrcPartCtlrController	::= {cwRsrcPartCtlrConfEntry 1}
cwRsrcPartCtlrRowStatus	::= {cwRsrcPartCtlrConfEntry 2}

The MIB objects are listed in Table 3-72.

Table 3-72 cwRsrcPartCtlrConf Table MIB Objects

Name	Syntax	Description	Default Value
cwRsrcPartCtlrConf Table	`Sequence of cwRsrcPartCtlrConfEntry`	Configures additional controllers to manage a resource partition. Max-Access: not-accessible Status: current	none
cwRsrcPartCtlrConf Entry	`CwRsrcPartCtlrConfEntry`	Provides an entry for additional controllers to control an existing resource partition. The resource partition, identified by indexes `ifIndex` and `cwRsrcPartID`, must exist in the `cwRsrcPartConfTable`. An entry from this table is deleted if the `cwRsrcPartID` associated entry from `cwRsrcPartConfTable` is deleted. Max-Access: not-accessible Status: current The indexes include: • `ifIndex` • `cwRsrcPartID` • `cwRsrcPartCtlrController`	none
cwRsrcPartCtlr Controller	`Unsigned32 (1.. 255)`	Specifies the controller identifier. One resource partition is controlled by multiple controllers. Users may want to configure redundant controllers. Max-Access: not-accessible Status: current	none
cwRsrcPartCtlrRow Status	`RowStatus`	Creates a row in the `cwRsrcPartCtlrConfTable`. A redundant controller is added by setting this object to `createAndGo(4)`. A row is deleted by setting this object to `destroy(6)`. Max-Access: read-create Status: current	none

Cisco WAN Resource Partition Conformance Information

One object identifier is listed in Table 3-73.

Table 3-73 Cisco WAN Resource Partition Conformance Object Identifier

Name	Object Identifier
cwRsrcPartMIBCompliance	::= {cwRsrcPartMIBCompliances 1}

One MIB object is listed in Table 3-74.

Table 3-74 Cisco WAN Resource Partition Conformance MIB Object

Name	Mandatory Groups	Description		Default Value
cwRsrcPartMIB Compliance	`cwRsrcPartMIBGroup`	Describes the compliance statement for the resource partition management group. Status: current Module: AXSM The mandatory groups include:		none
		Group	**Description**	
		`cwRsrcPartIlmiMIBGroup`	Implementation of this group is not mandatory for switches which do not support ILMI.	
		`cwRsrcPartMappingGroup`	Implementation of this group is not mandatory for switches which do not support redundant controllers.	

Cisco WAN Resource Partition Units of Conformance

The object identifiers are listed in Table 3-75.

Table 3-75 Cisco WAN Resource Partition Units of Conformance Object Identifiers

Name	Object Identifier
cwRsrcPartMIBGroup	::= {cwRsrcPartMIBGroups 1}
cwRsrcPartIlmiMIBGroup	::= {cwRsrcPartMIBGroups 2}
cwRsrcPartMappingGroup	::= {cwRsrcPartMIBGroups 3}

The MIB objects for units of conformance are listed in Table 3-76.

Table 3-76 Cisco WAN Resource Partition Units of Conformance MIB Objects

Name	Objects	Description	Default Value
cwRsrcPartMIBGroup	cwRsrcPartController, cwRsrcPartEgrGuarPctBwConf, cwRsrcPartEgrMaxPctBwConf, cwRsrcPartIngGuarPctBwConf, cwRsrcPartIngMaxPctBwConf, cwRsrcPartEgrPctBwAvail, cwRsrcPartIngPctBwAvail, cwRsrcPartEgrPctBwUsed, cwRsrcPartIngPctBwUsed, cwRsrcPartVpiLo, cwRsrcPartVpiHigh, cwRsrcPartVciLo, cwRsrcPartVciHigh, cwRsrcPartGuarCon, cwRsrcPartMaxCon, cwRsrcPartUsedCon, cwRsrcPartAvailCon, cwRsrcPartRowStatus	Relates to the resource partition group. Status: current	none
cwRsrcPartIlmiMIB Group	cwRsrcPartIlmiEnabled, cwRsrcPartSignallingVpi, cwRsrcPartSignallingVci, cwRsrcPartIlmiTrapEnable, cwRsrcPartIlmiEstablishConPollIntvl, cwRsrcPartIlmiCheckConPollIntvl, cwRsrcPartIlmiConPollInactFactor	Relates to the ILMI configuration on a resource partition. Status: current	none
cwRsrcPartMapping Group	cwRsrcPartCtlrRowStatus	Configures redundant controllers for resource partitions. Status: current	none

Cisco WAN ATM Connection

The Cisco WAN ATM Connection MIB, which is used for ATM switch connection management, configures and provisions the Switched Permanent Virtual Circuit (SPVC) on the ATM interface. The MIB objects reside in the CISCO-WAN-ATM-CONN-MIB.my file.

Note The Cisco WAN ATM Connection MIB is supported by AXSM, AXSM-E, RPM, and defined under the StrataCom Enterprise. RPM is also implemented in PXM45.

The Cisco WAN ATM connection MIB objects include:

- SPVC Provisioning Model
- Cisco WAN ATM Connection Textual Conventions
- Channel Configuration Group
- Channel State Group
- Channel Test Group
- Cisco WAN ATM Connection Conformance Information

SPVC Provisioning Model

To use this MIB module, it is essential to understand the provisioning model employed in the MGX 8800 Series products. This provisioning model is designed for the multiple types of services such as Frame Relay, circuit emulation, voice, and so forth.

Connection termination points are called endpoints. These endpoints derive their characteristics from the type of service they handle. Thus, Frame Relay endpoints are characterized by a Frame Relay Data-Link Connection Identifier (DLCI) and the committed frame rate. The endpoints can also terminate an ATM interface.

The connection between these endpoints is a ATM Switched Virtual Circuit (SVC). The routing of a connection within an ATM network is handled by a network controller, for example, Private Network-to-Network Interface (PNNI).

From a user perspective, the endpoint is identified either by physical entities (for example, node name, shelf, slot, or physical line), or logical entities (such as a DLCI in a Frame Relay module, a T1 slot number in a voice module, or a VPI/VCI in an ATM module).

The above identifiers call user addresses. There is yet another set of addresses that defines the endpoints within an ATM network. This is the address of the endpoint that is known to the network controller and hence called “network address.” The network address is assigned by the switch when a user provisions an endpoint. The network address is mandatory for routing a connection. The network address in this ATM network is a combination of an Network Service Access Point (NSAP) address (a 20 octet number uniquely identifying a logical interface) and a VPI/VCI within this interface.

In a switched network, there are always a calling and a called party. Likewise in the ATM network, one endpoint is responsible for setting up the connection (making a call). This endpoint is termed the master endpoint, the other one being the slave. To route a call from the master to slave, the master endpoint needs to know the network address of the slave endpoint. And this information must be conveyed when the user provisions a master endpoint.

The SPVC provisioning module imposes a sequence in provisioning a connection.

1. NMS adds the slave endpoint. The slave endpoint user address is specified during the add operation. The switch assigns a network address to this added endpoint.
2. NMS gets the slave network address from switch.
3. NMS adds the master endpoint. The master user address is specified, along with the slave network address.
4. If both the endpoints are added, SPVC provisioning is deemed successful. The routing of the connection takes place only after the master endpoint is added.

Cisco WAN ATM Connection Textual Conventions

The names of the textual conventions are specified in the object syntax.

Note Object identifiers are not applicable.

The textual conventions are listed in Table 3-77.

Table 3-77 Cisco WAN ATM Connection Textual Convention MIB Objects

Name	Description	Syntax	Default Value
CiscoAtmService Category	Specifies the ATM forum service categories. To be backward compatible with UNI3.1, two Constant Bit Rate (CBR) service types `cbr2` and `cbr3` are added. Available Bit Rate (ABR) ForeSight Service type is also supported. Refer to the ATM Forum Traffic Management Specification, Version 4.0, Section 4.5.4. Status: current	`INTEGER {cbr1(1), vbr1RT(2), vbr2RT(3), vbr3RT(4), vbr1nRT(5), vbr2nRT(6), vbr3nRT(7), ubr1(8), ubr2(9), abr(10), cbr2(11), cbr3(12)}`	none
CiscoWanLpbk Types	Defines the following loopback configurations for a connection: • `noLoopback` (no loopback or clear configured loopback) • `destructive` (loopback all cells, causing data disruption) • `nonDestructive` (loopback performed using OAM loopback cells, does not disrupt regular traffic) Status: current	`INTEGER {noLpbk(1), destructive(2), nonDestructive(3)}`	none
CiscoWanLpbk Dir	Defines the direction where the loop is effected. The loops include • `external`—Loop port traffic back to port. This is applicable only for destructive mode. • `internal`—Loop switches egress traffic back to switch. This is applicable only for destructive mode. • `forward`—Inject OAM loopback cells towards the switching fabric (ingress). This is applicable only for nondestructive mode. • `reverse`—Inject OAM loopback cells towards the port (egress). This is applicable only for nondestructive mode. Status: current	`INTEGER {external(1), internal(2), forward(3), reverse(4)}`	none
CiscoWanTest Status	Defines the loopback test status at an endpoint. Status: current	`INTEGER {noStatus (1), lpbkInProgress(2), lpbkSuccess(3), lpbkAbort(4), lpbkTimeOut(5), lpbkInEffect(6)}`	none
CiscoWanOper Status	Defines the operational status of an endpoint. Status: current	`INTEGER {operOk(1), operFail(2), adminDown(3)}`	none
CiscoWanNsap AtmAddress	Determines the ATM address used by the networking entity. The only address type presently supported is NSAP (20 octets). Status: current	`OCTET STRING (SIZE(20))`	none

Table 3-77 Cisco WAN ATM Connection Textual Convention MIB Objects (continued)

Name	Description	Syntax	Default Value
CiscoWanAlarm State	Defines the following alarms at an endpoint that include: • `aisRdi`—Endpoint receiving AIS or RDI cells in ingress or egress direction. • `conditioned`—Networking entity has forced the endpoint out of service. This could be attributed to either routing failure or to a maintenance operation initiated by the networking entity. • `interfaceFail`—Interface to which this connection belongs has failed. • `ccFail`—OAM continuity check between the connection and its peer endpoint has detected a failure. • `mismatch`—Connection exists in the SM database, but not in the network controller database. Status: current	`INTEGER {aisRdi (1), egrAisRdi(2), conditioned (4), interfaceFail(8), ccFail(16), mismatch (32), ingAbitFail(64)}`	none
CiscoWanXmt State	Defines the following transmit states of an endpoint: • `normal`—Endpoint transmitting normal traffic. • `sendingAIS`—Endpoint inhibits regular traffic. It sends AIS on egress. • `sendingRDI`—Endpoint inhibits regular traffic. It sends RDI on egress. Status: current	`INTEGER {normal(1), sendingAIS(2), sendingRDI(3)}`	none
CiscoWanRcv State	Defines the following receive states of an endpoint that include: • `normal`—Endpoint is receiving normal traffic. • `receivingRDI`—Endpoint receiving RDI, in either ingress or egress. • `receivingAIS`—Endpoint receiving AIS, in either ingress or egress. • `ccFailure`—Endpoint does not receive OAM CC cells. Status: current	`INTEGER {normal(1), receivingRDI(2), receivingAIS(3), ccFailure(4)}`	none
CiscoWanERS Confg	Defines the following configuration for Explicit Rate Stamping (ERS). • `none`—Disables the ERS on connection. • `enableIngress`—Enables ERS in the ingress direction only. • `enableEgress`—Enables ERS in the egress direction only. • `enableBoth`—Enables ERS in both directions. Status: current	`INTEGER {none(1), enableIngress(2), enableEgress(3), enableBoth(4)}`	none

Table 3-77 Cisco WAN ATM Connection Textual Convention MIB Objects (continued)

Name	Description	Syntax	Default Value
CiscoWanVSVD Confg	Defines the Virtual Source/Virtual Destination (VS/VD) configuration applicable to the following endpoints: • `vsvdOff`—Disables VS/VD. • `vsvdOn`—Enables VS/VD. • `switchDefault`—Uses default settings on switch. Status: current	`INTEGER {vsvdOff(1), vsvdOn(2), switchDefault(3)}`	none
CiscoWanAisIW	Defines an SPVC for the following endpoints for the AIS capability: • `e2eAisCapable`—Endpoint capable of detecting and generating `e2eAIS`. • `segAisCapable`—Endpoint capable of detecting and generating `segAIS`. Status: current	`INTEGER {e2eAisCapable(1), segAisCapable(2)}`	none
AbrRateFactors	Defines the rate factors to be used in increasing or decreasing the ABR cell rate. Status: current	`INTEGER {oneOver32768(1), oneOver16384(2), oneOver8192(3), oneOver4096(4), oneOver2048(5), oneOver1024(6), oneOver512(7), oneOver256(8), oneOver128(9), oneOver64(10), oneOver32(11), oneOver16(12), oneOver8(13), oneOver4(14), oneOver2(15), one(16)}`	none

Channel Configuration Group

The Channel Configuration Group contains all objects required for configuring a SPVC on a ATM Capable Service Module.

The object identifiers are listed in Table 3-78.

Table 3-78 Channel Configuration Group Object Identifiers

Name	Object Identifier
cwAtmChanCnfgTable	::= {cwAtmChanCnfg 1}
cwAtmChanCnfgEntry	::= {cwAtmChanCnfgTable 1}
cwaChanVpi	::= {cwAtmChanCnfgEntry 1}
cwaChanVci	::= {cwAtmChanCnfgEntry 2}
cwaChanServiceCategory	::= {cwAtmChanCnfgEntry 3}

Table 3-78 Channel Configuration Group Object Identifiers (continued)

Name	Object Identifier
cwaChanVpcFlag	::= {cwAtmChanCnfgEntry 4}
cwaChanIdentifier	::= {cwAtmChanCnfgEntry 5}
cwaChanUploadCounter	::= {cwAtmChanCnfgEntry 6}
cwaChanStatsEnable	::= {cwAtmChanCnfgEntry 7}
cwaChanCCEnable	::= {cwAtmChanCnfgEntry 8}
cwaChanLocalVpi	::= {cwAtmChanCnfgEntry 9}
cwaChanLocalVci	::= {cwAtmChanCnfgEntry 10}
cwaChanLocalNSAPAddr	::= {cwAtmChanCnfgEntry 11}
cwaChanRemoteVpi	::= {cwAtmChanCnfgEntry 12}
cwaChanRemoteVci	::= {cwAtmChanCnfgEntry 13}
cwaChanRemoteNSAPAddr	::= {cwAtmChanCnfgEntry 14}
cwaChanControllerId	::= {cwAtmChanCnfgEntry 15}
cwaChanRoutingMastership	::= {cwAtmChanCnfgEntry 16}
cwaChanMaxCost	::= {cwAtmChanCnfgEntry 17}
cwaChanReroute	::= {cwAtmChanCnfgEntry 18}
cwaChanFrameDiscard	::= {cwAtmChanCnfgEntry 19}
cwaChanOperStatus	::= {cwAtmChanCnfgEntry 20}
cwaChanPCR	::= {cwAtmChanCnfgEntry 21}
cwaChanMCR	::= {cwAtmChanCnfgEntry 22}
cwaChanSCR	::= {cwAtmChanCnfgEntry 23}
cwaChanCDV	::= {cwAtmChanCnfgEntry 24}
cwaChanCTD	::= {cwAtmChanCnfgEntry 25}
cwaChanMBS	::= {cwAtmChanCnfgEntry 26}
cwaChanCDVT	::= {cwAtmChanCnfgEntry 27}
cwaChanPercentUtil	::= {cwAtmChanCnfgEntry 28}
cwaChanRemotePCR	::= {cwAtmChanCnfgEntry 29}
cwaChanRemoteMCR	::= {cwAtmChanCnfgEntry 30}
cwaChanRemoteSCR	::= {cwAtmChanCnfgEntry 31}
cwaChanRemoteCDV	::= {cwAtmChanCnfgEntry 32}
cwaChanRemoteCTD	::= {cwAtmChanCnfgEntry 33}
cwaChanRemoteMBS	::= {cwAtmChanCnfgEntry 34}
cwaChanRemoteCDVT	::= {cwAtmChanCnfgEntry 35}
cwaChanRemotePercentUtil	::= {cwAtmChanCnfgEntry 36}
cwaChanAbrICR	::= {cwAtmChanCnfgEntry 37}
cwaChanAbrADTF	::= {cwAtmChanCnfgEntry 38}
cwaChanAbrRDF	::= {cwAtmChanCnfgEntry 39}

Table 3-78 Channel Configuration Group Object Identifiers (continued)

Name	Object Identifier
cwaChanAbrRIF	::= {cwAtmChanCnfgEntry 40}
cwaChanAbrNRM	::= {cwAtmChanCnfgEntry 41}
cwaChanAbrTRM	::= {cwAtmChanCnfgEntry 42}
cwaChanAbrCDF	::= {cwAtmChanCnfgEntry 43}
cwaChanAbrFRTT	::= {cwAtmChanCnfgEntry 44}
cwaChanAbrTBE	::= {cwAtmChanCnfgEntry 45}
cwaChanAbrERS	::= {cwAtmChanCnfgEntry 46}
cwaChanAbrVSVDEnable	::= {cwAtmChanCnfgEntry 47}
cwaChanRowStatus	::= {cwAtmChanCnfgEntry 48}
cwaChanIntAbrVSVD	::= {cwAtmChanCnfgEntry 49}
cwaChanExtAbrVSVD	::= {cwAtmChanCnfgEntry 50}
cwaChanAisIWCapability	::= {cwAtmChanCnfgEntry 51}
cwaChanCLR	::= {cwAtmChanCnfgEntry 52}
cwaChanRemoteCLR	::= {cwAtmChanCnfgEntry 53}
cwaChanOamSegEpEnable	::= { cwAtmChanCnfgEntry 54 }

The MIB objects are listed in Table 3-79.

Table 3-79 Channel Configuration Group MIB Objects

Name	Syntax	Description	Default Value
cwAtmChanCnfg Table	`Sequence Of CwAtmChanCnfgEntry`	Contains mandatory endpoint configuration for all SPVC connections. Most of the objects in this table are applicable to all provisioned endpoints for all service categories. The object `cwaChanAbrZZZ` provides additional configuration information that applies only to ABR endpoints. Max-Access: not-accessible Status: current	none
cwAtmChanCnfg Entry	`cwAtmChanCnfgEntry`	Contains an entry in the `cwAtmChanCnfgTable`. Each entry corresponds to a connection endpoint identified by an interface (`ifIndex`), VPI, and VCI. Max-Access: not-accessible Status: current The indexes include: • `ifIndex` • `cwaChanVpi` • `cwaChanVci`	none

Table 3-79 Channel Configuration Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cwaChanVpi	Unsigned32 (0..4095)	Determines the VPI value of a VP or VC connection. The cwaChanVpcFlag indicates if this is a VP/VC connection. Max-Access: not-accessible Status: current	none
cwaChanVci	Unsigned32 (0..65535)	The VCI value of VC connection. The cwaChanVpcFlag serves to distinguish if this is a VP/VC connection. For a VPC, the VCI is irrelevant and is set to a value of -2. Max-Access: not-accessible Status: current	none
cwaChanService Category	CiscoAtmService Category	Identifies the service type to which this connection belongs. The service type specified is one among the ATM forum service types and implicitly determines the configuration for GCRA. Max-Access: read-create Status: current	none
cwaChanVpcFlag	TruthValue	Identifies whether or not there is a VP/VC endpoint. When set to true(1), it indicates a VP endpoint. Max-Access: read-create Status: current	none
cwaChanIdentifier	Unsigned32 (0..4294967295)	Represents a resource at the switch that has been assigned for this connection (identified by cwaChanVpi and cwaChanVci). An example of this resource is the Logical Channel Number/Global Logical Channel Number (LCN/GLCN) used by the switch. Another example is a record number assigned for this connection in a database of records. When a connection gets deleted, this resource gets released into a free pool. When a new endpoint is added (different cwaChanVpi and cwaChanVci), the switch reassigns this resource. Thus, all connections in a switch would have a unique cwaChanIdentifier. Since the number of resources in the switch is a lesser subset compared to possible combinations of ifIndex, VPI, and VCI, this number serves as a quick reference index between the switch and the NMS. This is especially useful during the configuration upload of connections from the switch to the NMS. Max-Access: read-only Status: current	none

Table 3-79 Channel Configuration Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cwaChanUpload Counter	`Unsigned32 (0..4294967295)`	Tracks the number of configuration changes that happen on a `cwaChanIdentifier`. The upload counter is associated only with the `cwaChanIdentifier` and not with the connection itself. This counter is used by the NMS to determine if a connection configuration has been modified and requires an upload. This function is conventionally achieved by time stamping using a time-of-day clock. The upload counter is incremented when one of the following schemes takes place: • Assignment of connection to a `cwaChanIdentifier`. This happens when a connection is added and assigned this `cwaChanIdentifier`. • De-assignment of a connection from a `cwaChanIdentifier`. This happens when a connection is deleted and the `cwaChanIdentifier` resource is released. • When there is a configuration change completed to the connection that is associated with this `cwaChanIdentifier`. • In a new system, an unutilized resource (`cwaChanIdentifier`) has a counter value of 0. When a connection is added to this `cwaChanIdentifier`, the counter is incremented, and is further incremented for any of the above operations. When a connection is deleted, the value of this counter is incremented and preserved until a new connection gets associated with this `cwaChanIdentifier`. Max-Access: read-only Status: current	none
cwaChanStats Enable	`TruthValue`	Limits imposed by software or hardware implementations can restrict the amount of statistical data that can be maintained in a physical entity (for example, a service module). Therefore, there is a need to restrict statistics collection to a smaller subset. This object serves the purpose of enabling or disabling statistics collection on a per-connection basis. In implementations which do not have such limitations, this object is set to `true(1)` for all connections. Max-Access: read-create Status: current	`false`

Table 3-79 Channel Configuration Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cwaChanCCEnable	TruthValue	Enables or disables Continuity Check (CC) on a connection endpoint. When continuity check is enabled, the endpoint anticipates OAM CC cells from its peer endpoint. OAM CC cells are sent when the peer endpoint does not have traffic cells to send. If the connection is idle and this endpoint has not received OAM CC cells for a period of 3.5 +/- 0.5 seconds, it declares continuity failure. This object serves to administratively control the CC feature. Typical implementations (of this feature) can choose to ignore this control or impose other conditions to actually enable CC cell flow. However, if this object is set to false(2), this feature is disabled. Max-Access: read-create Status: current	false
cwaChanLocalVpi	Unsigned32 (0..4095)	Identifies the internal VPI assigned to a local endpoint by the switch. The cwaChanLocalVpi, cwaChanLocalVci, and the cwaChanLocalNSAPAddr form a unique identifier for the connection endpoint in the networking domain. Max-Access: read-only Status: current	none
cwaChanLocalVci	Unsigned32 (0..65535)	Identifies the internal VCI assigned to a local endpoint by the switch. The cwaChanLocalVpi, cwaChanLocalVci, and the cwaChanLocalNSAPAddr form a unique identifier for the connection endpoint in the networking domain. Max-Access: read-only Status: current	none
cwaChanLocal NSAPAddr	CiscoWanNsapAtm Address	Identifies the internal Network Service Access Point (NSAP) assigned to a local endpoint by the switch. The cwaChanLocalVpi, cwaChanLocalVci, and the cwaChanLocalNSAPAddr form a unique identifier for the connection endpoint in the networking domain. Max-Access: read-only Status: current	none
cwaChanRemote Vpi	Unsigned32 (0..4095)	Identifies the VPI of the peer endpoint. The cwaChanRemoteVpi, cwaChanRemoteVpi, and the cwaChanRemoteNSAPAddr identify the peer endpoint in the networking domain. Max-Access: read-create Status: current	none

Table 3-79 Channel Configuration Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cwaChanRemote Vci	`Unsigned32 (0..65535)`	Identifies the VCI of the peer endpoint. The `cwaChanRemoteVpi`, `cwaChanRemoteVpi`, and the `cwaChanRemoteNSAPAddr` identify the peer endpoint in the networking domain. Max-Access: read-create Status: current	none
cwaChanRemote NSAPAddr	`CiscoWanNsapAtm Address`	Identifies the VCI of the peer endpoint. The `cwaChanRemoteVpi`, `cwaChanRemoteVpi`, and the `cwaChanRemoteNSAPAddr` identify the peer endpoint in the networking domain. Max-Access: read-create Status: current	none
cwaChanController Id	`Unsigned32 (1..255)`	Associates an endpoint with a specific controller. Usually resource partitioning makes the association between a controller and a range of VPI-VCI. There can be switches where hard partitioning of VPI-VCI cannot be implemented. This object serves to tie a specific VPI-VCI to a controller. Max-Access: read-create Status: current	`2`
cwaChanRouting Mastership	`TruthValue`	Identifies this endpoint as the master endpoint of the connection if set to `true(1)`. After a master endpoint is added, the networking entity initiates routing of a PVC connection only. Mastership of a PVC cannot be changed. Once provisioned, this object can be set only during row creation. Max-Access: read-create Status: current	`false`
cwaChanMaxCost	`Unsigned32 (0..4294967295)`	Uses the networking entity to select a route based on the cost factor. The cost of the links and nodes are contained in a path that are aggregated and compared against this object. The networking entity attempts to find a path whose aggregate cost factor is less than the value of this object. If the parameter is set to 0, it indicates to the switch that only the best available route is chosen. If this object is set to a value of `0xFFFFFFFF(4294967295)`, the switch ignores this metric in making routing decisions. **Note** A connection is not allowed on the slave end point. Therefore, a connection is set only on the master end point. Max-Access: read-create Status: current	`'FFFFFFFF'h`

Table 3-79 Channel Configuration Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cwaChanReroute	TruthValue	Uses the administrator to trigger the rerouting of the connection. Rerouting takes effect when this object is set to true(1). When set to false(2), no action is taken. A Get operation on this object always returns false(2). When this object is set, no other object other than the RowStatus is set in the MIB. A reroute can be triggered only from the master endpoint. Any attempt to trigger reroute from the slave endpoint would result in a failure of the SET operation. Max-Access: read-create Status: current	false
cwaChanFrame Discard	TruthValue	Enables the frame discard feature at the endpoint when set to true(1). Max-Access: read-create Status: current	false
cwaChanOper Status	CiscoWanOperStatus	Reflects operational status of an endpoint. This object is set to operFail(2), if the connection is not routed, if the endpoint receives AIS/RDI, or if there is a CC failure. If the connection is administratively down, this object is set to adminDown(3). For a normal case, this object is set to operOk(1). Max-Access: read-only Status: current	none
cwaChanPCR	Unsigned32 (0..4294967295)	Defines the Peak Cell Rate (PCR) for the local to remote direction. Units: cps Max-Access: read-create Status: current	none
cwaChanMCR	Unsigned32 (0..4294967295)	Defines the Minimum Cell Rate (MCR) for the local to remote direction. Units: cps Max-Access: read-create Status: current	none
cwaChanSCR	Unsigned32 (0..4294967295)	Defines the Sustainable Cell Rate (SCR) for the local to remote direction. Units: cps Max-Access: read-create Status: current	none

Table 3-79 Channel Configuration Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cwaChanCDV	Unsigned32 (0..16777215)	Specifies the maximum tolerable cell delay variation in the local to remote direction. If this parameter is set to 0xFFFFFF (16777215), it indicates to the switch that this parameter does not have significance in the SPVC call setup. **Note** A connection is not allowed on the slave end point. Therefore, a connection is set only on the master end point. Units: microseconds Max-Access: read-create Status: current	'FFFFFF'h
cwaChanCTD	Unsigned32 (0..65535)	Specifies the maximum tolerable network transfer delay in the local to remote direction. If this parameter is set to 0xFFFFF (16777215), it indicates to the switch that this parameter does not have significance in the SPVC call setup. **Note** A connection is not allowed on the slave end point. Therefore, a connection is set only on the master end point. Units: milliseconds Max-Access: read-create Status: current	'FFFF'h
cwaChanMBS	Unsigned32 (0..5000000)	Specifies the Maximum Burst Size (MBS) used in the local to remote direction. Refer to *ATM Forum Traffic Management Specification,* Version 4.0, Annex C for more information. Units: cells Max-Access: read-create Status: current	none
cwaChanCDVT	Unsigned32 (0..4294967295)	Specifies the Cell Delay Variation Tolerance (CDVT) used in the local to remote direction. If this parameter is set to 0xFFFFFFFF(4294967295), it indicates to the switch that it can use a default value of CDVT. Refer to *ATM Forum Traffic Management Specification*, Version 4.0, Annex C for more information. Units: microseconds Max-Access: read-create Status: current	'FFFFFFFF'h

Table 3-79 Channel Configuration Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cwaChanPercent Util	`Unsigned32 (0..100)`	Provides a per-connection control for overbooking bandwidth. This is used in conjunction with the VSI policy while performing Connection Admission Control (CAC), which is applied for the local to remote direction. Max-Access: read-create Status: current	`100`
cwaChanRemote PCR	`Unsigned32 (0..4294967295)`	Specifies the PCR for the remote to local direction. Units: cps Max-Access: read-create Status: current	none
cwaChanRemote MCR	`Unsigned32 (0..4294967295)`	Specifies the MCR for the remote to local direction. Units: cps Max-Access: read-create Status: current	none
cwaChanRemote SCR	`Unsigned32 (0..4294967295)`	Specifies the SCR for the remote to local direction. Units: cps Max-Access: read-create Status: current	none
cwaChanRemote CDV	`Unsigned32 (0..16777215)`	Specifies the maximum tolerable cell delay variation for the remote to local direction. If this parameter is set to `0xFFFFFF(16777215)`, it indicates to the switch that this parameter does not have significance in SPVC call setup. **Note** A connection is not allowed on the slave end point. Therefore, a connection is set only on the master end point. Units: microseconds Max-Access: read-create Status: current	`'FFFFFF'h`
cwaChanRemote CTD	`Unsigned32 (0..65535)`	Specifies the maximum tolerable network transfer delay in the remote to local direction. If this parameter is set to `0xFFFF(65535)`, it indicates to the switch that this parameter does not have significance in SPVC call setup. **Note** A connection is not allowed on the slave end point. Therefore, a connection is set only on the master end point. Units: ms Max-Access: read-create Status: current	`'FFFF'h`

Table 3-79 Channel Configuration Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cwaChanRemote MBS	Unsigned32 (0..5000000)	Specifies the MBS used in the remote to local direction. Refer to *ATM Forum Traffic Management Specification*, Version 4.0, Annex C for more information. Units: cells Max-Access: read-create Status: current	none
cwaChanRemote CDVT	Unsigned32 (0..4294967295)	Specifies the CDVT is used in the remote to local direction. If this parameter is set to 0xFFFFFFFF(4294967295), it indicates the switch can use a default value of CDVT. Refer to *ATM Forum Traffic Management Specification*, Version 4.0, Annex C for more information. Units: cells Max-Access: read-create Status: current	'FFFFFFFF'h
cwaChanRemote PercentUtil	Unsigned32 (0..100)	Provides a per-connection control for overbooking bandwidth. This object is used in conjunction with the VSI interface policy while performing CAC, which is applied for the remote to local direction. Max-Access: read-create Status: current	100
cwaChanAbrICR	Unsigned32 (0..4294967295)	Specifies the Initial Cell Rate (ICR) or the rate at which a source can send after an idle period. The value must not be larger than the PCR. *Refer to ATM Forum Traffic Management Specification*, Version 4.0, Section 5.10.2 for more information. Units: cps Max-Access: read-create Status: current	none
cwaChanAbrADTF	Unsigned32 (1..1023)	Describes the ACR Decrease Time Factor. This is the time permitted between sending RM cells before the rate is decreased to the ICR. Refer to *ATM Forum Traffic Management Specification*, Version 4.0, Section 5.10.2 for more information. Units: 10 ms Max-Access: read-create Status: current	none

Table 3-79 Channel Configuration Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cwaChanAbrRDF	AbrRateFactors	Describes the Rate Decrease Factor. Controls the rate decrease which occurs when backward RM cells with CI=1 are received. Larger values lead to faster rate decrease. This factor is applied at both the local and remote endpoints, thus making the loop control symmetrical. Backward RM cells at the local endpoint correspond to RM cells received in the remote to local direction. Backward RM cells at the remote endpoint correspond to RM cells received in the local to remote direction. Refer to *ATM Forum Traffic Management Specification*, Version 4.0, Section 5.10.2 for more information. Max-Access: read-create Status: current	none
cwaChanAbrRIF	AbrRateFactors	Describes the Rate Increase Factor. Controls the rate increase which occurs when a backward RM cell is received with CI=0 and NI=0. Larger values lead to a faster rate increase. This factor is applied at both the local and remote endpoints, thus making the loop control symmetrical. Backward RM cells at the local endpoint correspond to RM cells received in the remote to local direction. Backward RM cells at the remote endpoint correspond to RM cells received in the local to remote direction. Refer to *ATM Forum Traffic Management Specification*, Version 4.0, Section 5.10.2, for more information. Max-Access: read-create Status: current	none
cwaChanAbrNRM	INTEGER {nrm2(1), nrm4(2), nrm8(3), nrm16(4), nrm32(5), nrm64(6), nrm128(7), nrm256(8)}	Determines the maximum number of cells a source can send for each forward RM cell. Refer to *ATM Forum Traffic Management Specification*, Version 4.0, Section 5.10.2 for more information. Max-Access: read-create Status: current	none
cwaChanAbrTRM	INTEGER {trm0point78125(1), trm1point5625(2), trm3point125(3), trm6point25(4), trm12point5(5), trm25(6), trm50(7), trm100(8)}	Specifies the upper limit on the time between forward RM cells for an active source (in ms). Refer to *ATM Forum Traffic Management Specification*, Version 4.0, Section 5.10.2, for more information. Max-Access: read-create Status: current	none

Table 3-79 Channel Configuration Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cwaChanAbrCDF	`INTEGER {cdf0(1), cdfOneOver64(2), cdfOneOver32(3), cdfOneOver16(4), cdfOneOver8(5), cdfOneOver4(6), cdfOneOver2(7), cdfOne(8)}`	Controls the rate decrease associated with lost or delayed backward RM cells for the Cutoff Decrease Factor. Larger values result in a faster rate decrease. Refer to *ATM Forum Traffic Management Specification*, Version 4.0, Section 5.10.2 for more information. Max-Access: read-create Status: current	none
cwaChanAbrFRTT	`Unsigned32 (0..16700000)`	Delays from the source to a destination network the fixed round-trip time or the sum of the fixed propagation. Refer to *ATM Forum Traffic Management Specification*, Version 4.0, Section 5.10.2, for more information. Units: microseconds Max-Access: read-create Status: current	none
cwaChanAbrTBE	`Unsigned32 (0..16777215)`	Describes the Transient Buffer Exposure. Before the first RM cell returns, the negotiated number of cells that the network would like to limit is the source to sending during startup periods. Refer to *ATM Forum Traffic Management Specification*, Version 4.0, Section 5.10.2, for more information. Max-Access: read-create Status: current	none
cwaChanAbrERS	`CiscoWanERSConfg`	Configures the endpoint for Explicit Rate Stamping. Refer to the textual convention for the values. Max-Access: read-create Status: current	none
cwaChanAbrVSVD Enable	`TruthValue`	Specifies that the ABR connections require close loop control to limit the transmission rate, which depends on the network bandwidth. This close loop is end-to-end or between intermediate network segments. When terminating an ABR VPL, the endpoint acts like a Virtual Destination to the incoming traffic and generate backward RM cells. While doing this, it also needs to act as a virtual source and send forward RM cells to the real destination. This feature can be enabled or disabled under the control of this object. When set to `true(1)`, this feature is enabled. Refer to *ATM Forum Traffic Management Specification*, Version 4.0, Section 5.10.2. Max-Access: read-create Status: current	`false`

Table 3-79 Channel Configuration Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cwaChanRow Status	RowStatus	Creates, modifies, or deletes an entry in the ciscoWanAtmChanTable. A row is created using the CreateAndGo option. When the row is successfully created, the RowStatus is set to active by the agent. A row is deleted by setting the RowStatus to destroy. When there is a need to administratively down the connection, the RowStatus is set to notInService. When the switch completes the down operation, the value of this object is notInService. The connection is made active again by setting this object to active. Administrative status control is limited to the master endpoint only. The switch rejects any request for admin state change on the slave endpoint. Other options such as CreateAndWait are not used. Max-Access: read-create Status: current	none
cwaChanIntAbr VSVD	CiscoWanVSVD Confg	Enables or disables the Virtual Source/Virtual Destination (VS/VD) internal to a segment; for example, the closed loop control is affected between the two provisioned endpoints of the SPVC. Refer to the *ATM Forum Traffic Management Specification*, Version 4.0, Section 5.10.2, for more information. Max-Access: read-create Status: current	switchDefault
cwaChanExtAbr VSVD	CiscoWanVSVD Confg	Enables or disables the VS/VD external to the segment that hosts the two endpoints of the SPVC; for example, the closed loop control is affected outside the segment towards the CPE or another segment. Refer to the *ATM Forum Traffic Management Specification*, Version 4.0, Section 5.10.2. Max-Access: read-create Status: current	switchDefault
cwaChanAisIW Capability	CiscoWanAisIW	Achieves OAM interoperability between switches that cannot generate or detect segment AIS cells. This attribute enables the new generation of switches to understand the OAM capability of the peer endpoint to generate or detect seg or e2eAIS as required. The value of this attribute is set during provisioning by network management. Max-Access: read-create Status: current	e2eAisCapable

Table 3-79 Channel Configuration Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cwaChanCLR	`Unsigned32 (1..15)`	Represents the maximum tolerable cell loss ratio for the encoded value in the local to remote direction. The actual CLR value is derived as the negative logarithm of this value. Max-Access: read-create Status: current	`6`
cwaChanRemote CLR	`Unsigned32 (1..15)`	Represents the maximum tolerable CLR for the value in the remote to local direction. The actual CLR value is derived as the negative logarithm of this value. Max-Access: read-create Status: current	`6`
cwaChanOamSeg EpEnable	`INTEGER { oamSegEp(1), nonOamSegEp(2)}`	Controls the setting and resetting of the OAM segment endpoint. When the `cwaChanOamSegEpEnable` object is set to `OamSegEp (1)`, the SPVC endpoint terminates all segment OAM FM cells, and loopbacks all OAM segment loopback cells. When the `cwaChanOamSegEpEnable` object is set to `NonOamSegEp (2)`, the SPVC endpoint is configured as a nonendpoint for OAM cells. All OAM Frequency Modulation (FM) cells and loopback cells pass transparently through the SPVC endpoint. If the `cwaChanOamSegEpEnable` object is not specified during a `Set` operation, the switch applies a default configuration maintained for the port. Max-Access: read-create Status: current	none

Channel State Group

The Channel State Group contains all objects reflecting the state of an SPVC on a ATM Capable Service Module.

The object identifiers are listed in Table 3-80.

Table 3-80 Channel State Group Object Identifiers

Name	Object Identifier
cwAtmChanStateTable	::= {cwAtmChanState 1}
cwAtmChanStateEntry	::= {cwAtmChanStateTable 1}
cwaChanAlarmState	::= {cwAtmChanStateEntry 1}
cwaChanEgressXmtState	::= {cwAtmChanStateEntry 2}
cwaChanEgressRcvState	::= {cwAtmChanStateEntry 3}
cwaChanIngressXmtState	::= {cwAtmChanStateEntry 4}
cwaChanIngressRcvState	::= {cwAtmChanStateEntry 5}

The MIB objects are listed in Table 3-81.

Table 3-81 Channel State Group MIB Objects

Name	Syntax	Description	Default Value
cwAtmChanStateTable	`Sequence of cwAtmChanStateEntry`	Contains the channel status information for all ATM connections configured in `cwAtmChanCnfgTable`. Max-Access: not-accessible Status: current	none
cwAtmChanStateEntry	`cwAtmChanStateEntry`	Provides an entry in `cwAtmChanStateTable`. Max-Access: not-accessible Status: current The indexes include: • `ifIndex` • `cwaChanVpi` • `cwaChanVci`	none
cwaChanAlarmState	`CiscoWanAlarmState`	Defines the alarms associated with an endpoint. Refer to the description in the textual convention. Max-Access: read-only Status: current	none
cwaChanEgressXmt State	`CiscoWanXmtState`	Specifies the state of the transmit portion of the endpoint in the egress direction. Max-Access: read-only Status: current	none
cwaChanEgressRcv State	`CiscoWanRcvState`	Specifies the state of the receive portion of the endpoint in the egress direction. Max-Access: read-only Status: current	none
cwaChanIngressXmt State	`CiscoWanXmtState`	Specifies the state of the transmit portion of the endpoint in the ingress direction. Max-Access: read-only Status: current	none
cwaChanIngressRcv State	`CiscoWanRcvState`	Specifies the state of the receive portion of the endpoint in the ingress direction. Max-Access: read-only Status: current	none

Channel Test Group

The Channel Test Group contains all the objects needed for SPVC diagnostics.

The object identifiers are listed in Table 3-82.

Table 3-82 Channel Test Group Object Identifiers

Name	Object Identifier
cwAtmChanTestTable	::= {cwAtmChanTest 1}
cwAtmChanTestEntry	::= {cwAtmChanTestTable 1}
cwaChanTestType	::= {cwAtmChanTestEntry 1}
cwaChanTestDir	::= {cwAtmChanTestEntry 2}
cwaChanTestIterations	::= {cwAtmChanTestEntry 3}
cwaChanTestState	::= {cwAtmChanTestEntry 4}
cwaChanTestRoundTripDelay	::= {cwAtmChanTestEntry 5}

The MIB objects are listed in Table 3-83.

Table 3-83 Channel Test Group MIB Objects

Name	Syntax	Description	Default Value
cwAtmChanTestTable	`Sequence of cwAtmChanTestEntry`	Contains configuration information for performing connection diagnostics on ATM connections and obtaining results after testing. Max-Access: not-accessible Status: current	none
cwAtmChanTestEntry	`cwAtmChanTestEntry`	Provides an entry in `cwAtmChanTestTable` for each ATM connection endpoints. Max-Access: not-accessible Status: current The indexes include: • `ifIndex` • `cwaChanVpi` • `cwaChanVci`	none

Table 3-83 Channel Test Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cwaChanTestType	CiscoWanLpbkTypes	Configures an endpoint in loopback. To configure a destructive loopback, set this object to destructive(2). The direction for this destructive loopback is specified using the cwaChanTestDir object. To deconfigure destructive loopback, this object needs to be set to noLpbk(1). To configure a nondestructive loopback, this object needs to be set to nonDestructive(3). The direction for this nondestructive loopback is specified using the cwaChanTestDir object. Nondestructive loopback gets deconfigured automatically at the switch after a certain timeout. However, if there is a need to abort a nondestructive loopback, a Set operation on this object should be performed with a value noLpbk (1). Attempting to set a channel in loopback when one is already in progress results in a failure of the Set operation (with appropriate error code). Max-Access: read-create Status: current	noLpbk
cwaChanTestDir	CiscoWanLpbkDir	Specifies the direction in which loopback is effected. For destructive loopback, this takes values external(1) and internal(2). For nondestructive loopback, this takes values forward(3) and reverse(4). When cwaChanTestType is noLpbk(1), this object is ignored. Max-Access: read-create Status: current	none
cwaChanTestIterations	Unsigned32 (1..255)	Specifies the number of loopback cells to inject. This applies to the nondestructive mode of loopback tests only. The Get operation on this object returns the successful number of loopback iterations performed on the endpoint. Max-Access: read-create Status: current	1

Table 3-83 Channel Test Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cwaChanTestState	`CiscoWanTestStatus`	Reflects the status of the last requested test. If a loopback had never been effected on an endpoint, this object reads `noStatus(1)`. If a destructive mode of loopback is configured on an endpoint, this object always reads `lpbkInEffect(6)`. If a nondestructive mode of loopback is in progress, this value reads `lpbkInProgress(2)`. If a nondestructive mode of loopback has been completed successfully, this reads `lpbkSuccess(3)`. If a nondestructive mode of loopback has aborted, either due to user request or switch's limitation, this reads `lpbkAbort(4)`. If a nondestructive mode of loopback has failed, this reads `lpbkTimeOut(5)`. Max-Access: read-only Status: current	`noStatus`
cwaChanTestRound TripDelay	`Unsigned32 (1..100000000)`	Specifies that this object is applicable only for the nondestructive mode of loopback. It returns the round trip delay measured during the last nondestructive loopback test. Units: microseconds Max-Access: read-only Status: current	none

Cisco WAN ATM Connection Conformance Information

One object identifier is listed in Table 3-84.

Table 3-84 Cisco WAN ATM Connection Conformance Object Identifier

Name	Object Identifier
cwaChanAbrVSVDEnable	::= {ciscoWanAtmConnMIBCompliances 1}

The MIB objects are listed in Table 3-85.

Table 3-85 Cisco WAN ATM Connection Conformance MIB Objects

Name	Syntax	Description	Default Value
ciscoWanAtmConnMIB Compliance	none	Specifies the compliance statement for the Cisco ATM connection management group. **Note** Implementation of the `ciscoWanAtmConnMIBCompliance` group is optional for the systems where connection state is not available on a per-connection basis. Status: current Module: AXSM `ciscoWanAtmConnChanMIBGroup` is the mandatory group.	none
cwaChanFrameDiscard	none	Specifies the read and write access is not required for those switches that do not support the frame discard feature. Min-Access: not-accessible	none
cwaChanCCEnable	none	Specifies the read and write access is not required for those switches that do not support Continuity Check capability. Min-Access: not-accessible	none
cwaChanStatsEnable	none	Specifies the read and write access is not required for those switches that do not support the connection related statistics collection. Min-Access: not-accessible	none
cwaChanAbrERS	none	Specifies the read and write access is not required for those switches that do not support Explicit Rate Stamping (ERS). Min-Access: not-accessible	none
cwaChanAbrVSVDEnable	none	Specifies that the read and write access is not required for those switches that do not support the VS/VD feature. Min-Access: not-accessible	none

Cisco WAN ATM Connection Units of Conformance

The object identifiers are listed in Table 3-86.

Table 3-86 Cisco WAN ATM Connection Units of Conformance Object Identifiers

Name	Object Identifier
ciscoWanAtmConnChanMIBGroup	::= {ciscoWanAtmConnMIBGroups 1}
ciscoWanAtmConnStateGroup	::= {ciscoWanAtmConnMIBGroups 2}

The MIB objects are listed in Table 3-87.

Table 3-87 Cisco WAN ATM Connection Units of Conformance MIB Objects

Name	Objects	Description	Default Value
ciscoWanAtmConnChanMIBGroup	cwaChanServiceCategory, cwaChanVpcFlag, cwaChanStatsEnable, cwaChanCCEnable, cwaChanUploadCounter, cwaChanIdentifier, cwaChanLocalVpi, cwaChanLocalVci, cwaChanLocalNSAPAddr, cwaChanRemoteVpi, cwaChanRemoteVci, cwaChanRemoteNSAPAddr, cwaChanControllerId, cwaChanRoutingMastership, cwaChanMaxCost, cwaChanReroute, cwaChanFrameDiscard, cwaChanOperStatus, cwaChanPCR, cwaChanMCR, cwaChanSCR, cwaChanCDV, cwaChanCTD, cwaChanMBS, cwaChanCDVT, cwaChanPercentUtil, cwaChanRemotePCR, cwaChanRemoteMCR, cwaChanRemoteSCR,cwaChanRemoteCDV, cwaChanRemoteCTD, cwaChanRemoteMBS, cwaChanRemoteCDVT, cwaChanRemotePercentUtil, cwaChanAbrICR, cwaChanAbrADTF, cwaChanAbrRDF, cwaChanAbrRIF, cwaChanAbrNRM, cwaChanAbrTRM, cwaChanAbrCDF, cwaChanAbrFRTT, cwaChanAbrTBE, cwaChanAbrERS, cwaChanAbrVSVDEnable, cwaChanRowStatus, cwaChanIntAbrVSVD, cwaChanExtAbrVSVD, cwaChanAisIWCapability, cwaChanCLR, cwaChanRemoteCLR,cwaChanTestType, cwaChanTestDir, cwaChanTestIterations, cwaChanTestState, cwaChanTestRoundTripDelay	Specifies objects related to connection information. Status: current	none
ciscoWanAtmConnStateGroup	cwaChanAlarmState, cwaChanEgressXmtState, cwaChanEgressRcvState, cwaChanIngressXmtState, cwaChanIngressRcvState	Specifies objects that are related to the connection state information. Status: current	none

Cisco WAN Feeder

This section describes the configuration of a port for an ATM Switch Card, which recognizes the information for the feeder connection. The Cisco WAN Feeder MIB objects reside in the CISCO-WAN-FEEDER-MIB.my file.

Note The Cisco WAN Feeder MIB is not supported by AXSM-E.

The feeder is an external ATM switch, which is connected to an ATM switch that allows PNNI to provide Local Management Interface (LMI). LMI provides a set of enhancements to a Frame Relay specification for managing complex internetworks. LMI extensions include global addressing, virtual-circuit status messages, and multicasting. However, Extended Local Management Interface (XLMI) is an LMI type that implements both Automatic Routing Management and PNNI. Automatic Routing Management provides connection management for the network with only Cisco nodes.

Note The Cisco WAN Feeder MIB is supported by AXSM and defined under the StrataCom Enterprise.

The Cisco WAN Feeder MIB objects include:

- Feeder Table
- Cisco WAN Feeder Conformance Information

Feeder Table

The object identifiers are listed in Table 3-88.

Table 3-88 Feeder Table Object Identifiers

Name	Object Identifier
cwfFeederTable	::= { cwfFeeder 1 }
cwfFeederEntry	::= { cwfFeederTable 1 }
cwfFeederIfNum	::= { cwfFeederEntry 1 }
cwfFeederName	::= { cwfFeederEntry 2 }
cwfLanIP	::= { cwfFeederEntry 3 }
cwfNetIP	::= { cwfFeederEntry 4 }
cwfRemoteShelf	::= { cwfFeederEntry 5 }
cwfRemoteSlot	::= { cwfFeederEntry 6 }
cwfRemotePort	::= { cwfFeederEntry 7 }
cwfFeederType	::= { cwfFeederEntry 8 }
cwfModelNumber	::= { cwfFeederEntry 9 }
cwfLMIAdminStatus	::= { cwfFeederEntry 10 }
cwfLMIOperStatus	::= { cwfFeederEntry 11 }
cwfFeederNodeAlarm	::= { cwfFeederEntry 12 }

Table 3-88 Feeder Table Object Identifiers (continued)

Name	Object Identifier
cwfFeederRowStatus	::= { cwfFeederEntry 13 }
cwfLMIType	::= { cwfFeederEntry 14 }

Table 3-89 lists the MIB objects.

Table 3-89 Feeder Table MIB Objects

Name	Syntax	Description	Default Value
cwfFeederTable	`SEQUENCE OF CwfFeederEntry`	Contains the entries for the feeders and is used to add or delete feeders to support LMI. The information in these entries is advertised to LMI, which uses a dependent system for implementation when an entry is created or activated. Max-Access: not-accessible Status: current	none
cwfFeederEntry	`CwfFeederEntry`	Contains an entry in the `cwfFeederTable`. The entries in this table are created by setting the `cwfFeederRowStatus` object to `createAndGo(4)`. The entries in this table are deleted by setting the `cwfFeederRowStatus` object to `destroy(6)`. Also, the entries are created or deleted through the CLI. Max-Access: not-accessible Status: current The index is `cwfFeederIfNum`.	none
cwfFeederIfNum	`InterfaceIndex`	Specifies the unique interface number of the ATM virtual interface. The value of this object is equal to the MIB II `ifIndex` value of the ATM virtual interface, which is identified by the `ifType` value `atmVirtual(149)`. Max-Access: not-accessible Status: current	none
cwfFeederName	`DisplayString`	Provides the name of the feeder. If the feeder name is not available, this object contains the Octet string of length 0. Max-Access: read-only Status: current	none
cwfLanIP	`IpAddress`	Provides the LAN IP address of the feeder. This IP address is used for the Ethernet interface. Max-Access: read-only Status: current	none
cwfNetIP	`IpAddress`	Provides the Network IP address of the feeder that is used for the ATM interface. Max-Access: read-only Status: current	none

Table 3-89 Feeder Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwfRemoteShelf	`Integer32 (1..4)`	Provides the remote shelf number of the feeder module. Max-Access: read-only Status: current	none
cwfRemoteSlot	`Integer32 (1..32)`	Provides the remote slot number of the feeder module. Max-Access: read-only Status: current	none
cwfRemotePort	`Integer32 (1..64)`	Provides the remote physical port (line) number of the feeder module. Max-Access: read-only Status: current	none
cwfFeederType	`INTEGER {fdrIPX(1), fdrBPX(2), fdrIpxAF(3), fdrBASIS(4), fdrUNKNOWN(5), fdrUNI(6), fdrAPS(7), fdrIGX(8), fdrIgxAF(9), fdrVSI(10), fdrPAR(11), fdrNON(12)}`	Identifies the feeder type. When `cwfLMIType` is `xLMI(2)`, `fdrNON(12)` is applicable. Other values are applicable for `cwfLMIType = feeder(1)`. The values include: • `fdrIPX`—Feeder is an IPX node in a routing network. • `fdrBPX`—Feeder is a BPX node in a routing network. • `fdrIpxAF`—Feeder is a standalone IPX node. • `fdrBASIS`—Feeder is a standalone BASIS node. • `fdrUNKNOWN`—Feeder is unknown. • `fdrUNI`—Feeder is a UNI AIT (phase 0). • `fdrAPS`—Feeder is an APS (Adjunct Processor Shelf). • `fdrIGX`—Feeder is an IGX node in a routing network. • `fdrIgxAF`—Feeder is a standalone IGX node. • `fdrVSI`—Feeder is a VSI Controller. • `fdrPAR`—Feeder is a PAR. • `fdrNON`—This value is a nonfeeder type. Max-Access: read-only Status: current	none
cwfModelNumber	`Integer32 (1..2147483647)`	Identifies the feeder model number. Max-Access: read-only Status: current	none
cwfLMIAdmin Status	`INTEGER { up(1), down(2) }`	Provides the feeder LMI administration state configuration capabilities for the desired state of the interface. The value for this object is ignored during row creation. Max-Access: read-create Status: current	`up`

Table 3-89 Feeder Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwfLMIOper Status	`INTEGER { up(1), down(2) }`	Provides the operational state of the LMI connection. Max-Access: read-only Status: current	none
cwfFeederNode Alarm	`INTEGER { clear (1), minor (2), major (3), critical (4), unknown (5) }`	Identifies the feeder node alarm status and provides the alarm status in hierarchical order. If there are no feeder node alarms, it is signified with none. Minor alarms, with no major alarms, are indicated with minor. If the node has one or more major alarms, regardless of minor alarm status, the alarm is indicated by major. Max-Access: read-only Status: current	none
cwfFeederRow Status	`RowStatus`	Adds or deletes the feeder entry. The row is created by setting this object to `createAndGo(4)`. The row is deleted by setting this object to `destroy(6)`. **Note** This object does not support any other values. Max-Access: read-create Status: current	none
cwfLMIType	`INTEGER { feeder(1), xLMI(2)}`	Identifies the LMI type used for the network to connect to the LMI port. Once created, `cwfLMIType` cannot be modified. Max-Access: read-create Status: current	`feeder`

Cisco WAN Feeder Conformance Information

One object identifier is listed in Table 3-90.

Table 3-90 Cisco WAN Feeder Conformance Object Identifier

Name	Object Identifier
cwfMIBCompliance	::= {cwfMIBCompliances 1}

One MIB object is listed in Table 3-91.

Table 3-91 Cisco WAN Feeder Conformance MIB Object

Name	Mandatory Group	Description	Default Value
cwfMIBCompliance	`cwfFeederGroup`	Specifies the compliance statement for the Cisco Feeder Group. Status: current Module: AXSM	none

Cisco WAN Feeder Units of Conformance

One object identifier is listed in Table 3-92.

Table 3-92 Cisco WAN Feeder Units of Conformance Object Identifier

Name	Object Identifier
cwfFeederGroup	::= { cwfMIBGroups 1}

One MIB object is listed in Table 3-93.

Table 3-93 Cisco WAN Feeder Units of Conformance MIB Object

Name	Objects	Description	Default Value
cwfFeederGroup	`cwfFeederName, cwfLanIP, cwfNetIP, cwfRemoteShelf, cwfRemoteSlot, cwfRemotePort, cwfFeederType, cwfModelNumber, cwfLMIAdminStatus, cwfLMIOperStatus, cwfFeederNodeAlarm, cwfFeederRowStatus`	Describes the objects related to configuring a feeder. Status: current	none

Cisco WAN Module

This section describes the MIB objects to configure connection specific parameters and statistics information in a Service Module (SM). SM is defined as any module which provides services such as ATM, Frame Relay, or Voice in a WAN switch. The Cisco WAN module MIB resides in the CISCO-WAN-MODULE-MIB.my file.

Note The Cisco WAN Module MIB is supported by AXSM and AXSM-E.

The Cisco WAN module MIB objects include:

- Cisco WAN Module Textual Convention
- Module Configuration Table
- Statistics Configuration Table

Cisco WAN Module Textual Convention

The names of the textual conventions are specified in the object syntax.

One MIB object is listed in Table 3-94.

Note Textual conventions do not contain object identifiers.

Table 3-94 Cisco WAN Module Textual Convention MIB Object

Name	Description	Syntax	Default Value
StatisticsLevel	Configures the statistics level for the service module. The statistic level on a module dictates the type and amount of statistics to be collected. The a predefined set of statistics are associated with each level and are enabled when a level is set. Also, setting the statistics level to *N* enables counters that are associated with levels *N*-1 through 1. The following are the valid values: • notApplicable(0)—Specifies that the statistics level is not applicable. • levelOne(1)—Enables the level one counters. • levelTwo(2)—Enables level two and level one counters. • levelThree(3)—Enables level one, level two, and level three counters. Status: current	INTEGER {notApplicable(0), levelOne(1), levelTwo(2), levelThree(3 }	none

Module Configuration Table

The object identifiers are listed in Table 3-95.

Table 3-95 Module Configuration Table Object Identifiers

Name	Object Identifier
cwmConfigTable	::= { cwmConfig 1}
cwmConfigEntry	::= { cwmConfigTable 1 }
cwmIndex	::= {cwmConfigEntry 1 }

Table 3-95 Module Configuration Table Object Identifiers (continued)

Name	Object Identifier
cwmIngressSCTFileId	::= {cwmConfigEntry 2 }
cwmIngressSCTFileName	::= { cwmConfigEntry 3 }
cwmAutoLineDiagEnable	::= { cwmConfigEntry 4 }

The MIB objects are listed in Table 3-96.

Table 3-96 Module Configuration Table MIB Objects

Name	Syntax	Description	Default Value
cwmConfigTable	`SEQUENCE OF CwmConfigEntry`	Contains the objects required for configuring module specific parameters. These parameters are related to the hardware specific parameters, which can affect the ATM Connection characteristics. Max-Access: not-accessible Status: current	none
cwmConfigEntry	`CwmConfigEntry`	Provides an entry that contains information for each module. Max-Access: not-accessible Status: current The index is `cwmIndex`.	none
cwmIndex	`Unsigned32 (1..2147483647)`	Specifies the value that corresponds to the slot number where the module resides. However, system wide uniqueness is the only true requirement. Max-Access: not-accessible Status: current	none

Table 3-96 Module Configuration Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwmIngressSCT FileId	`Unsigned32 (0..255)`	Defines the file ID of the SCT file. The SCT holds the connection specific parameters for this module in the ingress direction (for example, network to switch). The SCT provides a means for inferring extended parameters, which are generally platform-specific, from the set of standard ATM protocol parameters passed in VSI connection setup primitives. A set of Service Templates are available in nonvolatile storage, and they are downloaded onto each SM on power up. SCT contains the following classes of data: • One class consists of parameters necessary to establish a Virtual Connection (VC), for example, per-VC, and includes entries such as Usage Parameter Control (UPC) actions, various bandwidth-related items, per-VC thresholds, and some hardware-specific items. • The second class of data items includes those necessary to configure the associated Class-of-Service Buffers that provide the Quality of Service (QoS) support. Max-Access: read-write Status: current	none
cwmIngressSCT FileName	`DisplayString`	Contains the SCT filename. Write access is not required. The filename has the `cwmIngressSCTFileId` value as the suffix. For example, the value `SCT.INGR.13` for this object indicates that 13 is the value of `cwmIngressSCTFileId`. Max-Access: read-write Status: current	none
cwmAutoLineDiag Enable	`INTEGER { enable(1), disable(2) }`	Enables and disables automatic diagnostic feature of physical lines on module. In case of a line alarm, enabling this feature temporarily suspends traffic in both directions and starts local loopback testing. Disabling this feature will not affect the traffic in one direction while another direction is in alarm. Max-Access: read-write Status: current	`disable`

Statistics Configuration Table

The object identifiers are listed in Table 3-97.

Table 3-97 Statistics Configuration Table Object Identifiers

Name	Object Identifier
cwmStatConfigTable	::= { cwmStatsConfig 1 }
cwmStatConfigEntry	::= { cwmStatConfigTable 1 }
cwmStatBucketInterval	::= {cwmStatConfigEntry 1 }
cwmStatCollectionInterval	::= {cwmStatConfigEntry 2 }
cwmStatCollectionStatus	::= {cwmStatConfigEntry 3 }
cwmStatCurrentLevel	::= {cwmStatConfigEntry 4 }
cwmStatLevelConfigured	::= {cwmStatConfigEntry 5 }
cwmStatMaximumConnections	::= { cwmStatConfigEntry 6 }

The MIB objects are listed in Table 3-98.

Table 3-98 Statistics Configuration Table MIB Objects

Name	Syntax	Description	Default Value
cwmStatConfig Table	`SEQUENCE OF CwmStatConfigEntry`	Specifies the objects that are required for configuring module statistic collection-related parameters. The statistics related to the interfaces and ATM connections are stored in a file. The statistics file is uploaded by the NMS applications using file transfer protocols, for example Trivial File Transfer Protocol (TFTP) or File Transfer Protocol (FTP). While current interval of data is being collected in memory, the previous interval of data is uploaded by NMS. Each interval data can overwrite the previous interval data in the statistic file after the interval is over. Max-Access: not-accessible Status: current	none
cwmStatConfig Entry	`CwmStatConfigEntry`	Provides the entry that contains statistics configuration information for the module. Max-Access: not-accessible Status: current The index is `cwmIndex`.	none

Table 3-98 Statistics Configuration Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwmStatBucket Interval	`INTEGER { five(5), ten(10), fifteen(15), twenty(20), thirty(30), sixty(60) }`	Contains the bucket interval (in minutes) that is used in collecting statistics. This specifies the interval is over the module that accumulates a sample. The value also defines the amount of time available to the NMS application to upload the statistic file, so that NMS does not miss one interval worth of data. Units: minutes Max-Access: read-write Status: current	`fifteen`
cwmStatCollection Interval	`INTEGER { default(0), one(1), five(5) }`	Describes the collection interval of statistics. Within a sampling interval, as defined by `cwmStatBucketInterval`, statistics counters are updated for every collection interval. The default value of the collection interval is the same as the bucket interval. For example, if the bucket interval is five minutes, `default(0)` and `five(5)` values for this object have the same effect. Units: minutes Max-Access: read-write Status: current	`default`
cwmStatCollection Status	`INTEGER { enable(1), disable(2) }`	Enables or disables the collection of statistics on the module. Max-Access: read-write Status: current	`enable`
cwmStatCurrent Level	`StatisticsLevel`	Describes the current statistics level of the module. A change in the card statistic level can take place only after a module reset. This object shows the current module statistic level. While the `cwmStatLevelConfigured` object is used to configure the module statistic level, the value set for the `cwmStatLevelConfigured` object takes affect after the next module reset. Max-Access: read-only Status: current	none

Table 3-98 Statistics Configuration Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwmStatLevel Configured	`StatisticsLevel`	Configures the statistics level for the module. The statistics level value set in this object takes affect only on reset of the module. This object is set only if `cwmStatCollectionStatus` is set to `enable(1)`. This object cannot be set to `notApplicable(0)`. Max-Access: read-write Status: current	`levelOne`
cwmStatMaximum Connections	`Unsigned32`	Determines the maximum number of connections for which the statistics are being collected. Max-Access: read-only Status: current	none

Cisco Bulk File

This section describes the MIB objects that make up the Cisco Bulk File which resides in the CISCO-BULK-FILE-MIB.my file. The bulk file MIB group creates and deletes SNMP data bulk files used for data transfer.

Note The Cisco Bulk File MIB is supported by AXSM, PXM45, and RPM, which is also implemented in PXM45.

The Cisco Bulk File MIB objects include:

- Overview
- File Definition Table
- File Object Table
- File Table
- Cisco Bulk File Conformance and Compliance Information

Overview

The file contains two types of fields: tags and data. Tags identify portions of the file. All other information is in the data fields.

Note For efficiency and compactness, data fields are not tagged with a type.

The interpreter of the data must know or have access to the appropriate MIB syntax descriptions to understand the file.

All data fields are positioned to a tag and every data field has a length prefix. All initial length prefixes are one byte. For any data type, the distinguished length value 255 indicates that the data content is NULL, which no data content value is available and no additional bytes are in the data field.

The following are the value definitions used for the Cisco Bulk File:

- INTEGER—Includes all data that maps to ASN.1 INTEGER, regardless of length, and whether the fields are signed or unsigned.

  The fields have a length prefix value of 0 to 8 followed by the bytes of data, high-order byte first. High order bytes that are all zero are omitted, thus, a length of zero indicates a value of zero. For signed numbers, leading bytes of all ones (hex FF) are omitted if the next remaining byte has the high bit. This implies that the file parser must know the difference between signed and unsigned integers.

- OCTET STRING—Determines a length prefix value of 0 to 2 for a subsequent unsigned byte count for the number of bytes in the OCTET STRING itself, which immediately follows the byte count. The byte count can range from 0 to 65,535.

- OBJECT IDENTIFIER—Determines a length of 0 to 128 used for the number of subidentifiers. Each subsequent subidentifier is encoded as an unsigned INTEGER of 0 to 4 bytes.

The following are the three bulk transfer file formats:

- ASN.1/BER Variable Bindings—Specifies the standard Bit Error Rate (BER) similar to the varbinds section of a response PDU.
- Bulk Binary—Specifies the binary form designed for fast, sequential processing and minimum redundancy.
- Bulk ASCII—Specifies the binary form, mechanically translated to human-readable ASCII.

ASN.1 and Bit Error Rate Variable Bindings Format

The ASN.1/BER format is identical to SNMP variable bindings where each object has a full OID and a fully tagged value. The file content is similar to a GetBulkRequest Protocol Data Unit (PDU) except that it does not use a large amount of uninstantiated values. In essence, the file contains no data at all for scalars or columns that cannot be read.

Bulk Binary Format

The bulk binary file layout directly reflects the contents of the cbfDefineFileObjectTable object. It has tagged sections corresponding to the cbfDefineObjectClass object with a few additional tags for utility purposes.

A tag is one byte with one of the applicable values listed in Table 3-99.

Table 3-99 Bytes Used for the Bulk Binary Format

Byte	Value
-2	row
-1	prefix
0	reserved
1	object
2	table

The prefix tag changes the default OID prefix that preceded all OIDs that are not MIB object data values. The prefix tag can appear anywhere another tag could appear. A prefix tag is followed by one OID data field. The default prefix is 1.3.6.1. A file is not set to the prefix to the default value.

Note When changing the prefix, the default portion must be included at the beginning of the new prefix.

Typically the prefix will change for each table or group of scalar objects.

An object tag is followed by one OID data field and one data field appropriate to the syntax of the object. This OID is the full OID for the object minus the current prefix.

A table tag is followed by one INTEGER data field whose value is the number of columns in the table as implemented by the agent. This is followed by one OID data field for each column. This is the OID for the column minus the prefix and the instance, for example, one subidentifier.

The OIDs are followed by one row for each row in the table. A row starts with a row tag and one OID data field containing only the instance portion of the OIDs for the objects in that row. Following this is one data field of appropriate type for each column.

Bulk ASCII Format

The bulk ASCII form mechanically translates bulk binary into human-readable text.

The indicator for a NULL value is a ~ character.

The following are the field definitions used for the bulk ASCII format:

- INTEGER—Specifies the integer value with a preceding - character for negative values and no leading zeros.
- OCTET STRING—Specifies the byte values in hexadecimal, lower case, two characters per byte, for example, with leading zeros, and no delimiters between bytes.
- OBJECT IDENTIFIER—Specifies the dotted decimal format.

A tag becomes the tag name, spelled out fully in lower case, followed by one blank and the data field(s) for the tag, separated by spaces, and ending with a carriage return or line feed. All tags are at the beginning of a line, which is terminated with a carriage return or line feed that immediately precedes the next tag or the end of the file.

File Definition Table

The object identifiers used for the file definition table are listed in Table 3-100.

Table 3-100 File Definition Table Object Identifiers

Name	Object Identifier
cbfDefineFileTable	::= { cbfDefine 9 }
cbfDefineFileEntry	::= { cbfDefineFileTable 1 }
cbfDefineFileIndex	::= { cbfDefineFileEntry 1 }
cbfDefineFileName	::= { cbfDefineFileEntry 2 }
cbfDefineFileStorage	::= { cbfDefineFileEntry 3 }
cbfDefineFileFormat	::= { cbfDefineFileEntry 4 }
cbfDefineFileNow	::= { cbfDefineFileEntry 5}
cbfDefineFileEntryStatus	::= { cbfDefineFileEntry 6 }

The MIB objects used for the file definition table are listed in Table 3-101.

Table 3-101 File Definition Table MIB Objects

Name	Syntax	Description	Default Value
cbfDefineFileTable	SEQUENCE OF CbfDefineFileEntry	Provides a table of bulk file definitions and creation controls. Max-Access: not-accessible Status: current	none
cbfDefineFileEntry	CbfDefineFileEntry	Creates the information for the bulk file. To create a bulk file, an application creates an entry in this table and to correspond to the entries in the cbfDefineObjectTable. When the entry in this table and the corresponding entries in the cbfDefineObjectTable are active, the application uses the cbfDefineFileNow object to create the file to correspond to an entry in the cbfStatusFileTable. To delete an entry in the cbfDefineFileTable, you ned to delete all the corresponding entries in the cbfDefineObjectTable and cbfStatusFileTable. Entries cannot be modified or deleted while the cbfDefineFileNow object is using the value running. Rows cannot be created without explicitly setting the cbfDefinefileEntryStatus object to either createAndGo or createAndWait. Max-Access: not-accessible Status: current The index is cbfDefineFileIndex.	none
cbfDefineFileIndex	Unsigned32 (1..4294967295)	Identifies this entry to an arbitrary integer. To create an entry, a management application can pick a random number. Max-Access: not-accessible Status: current	none
cbfDefineFileName	DisplayString (SIZE (1..255))	Creates the file name which is to be created. The explicit device or path in the value of this object can override the cbfDefineFileStorage object. Max-Access: read-create Status: current	none

Table 3-101 File Definition Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cbfDefineFileStorage	`INTEGER { ephemeral(1), volatile(2), permanent(3) }`	Determines the type of file storage. The following are the definitions of the values: • `ephemeral`—Specifies the data that exists in small amounts until read. An `ephemeral` file is read one at a time. • `volatile`—Specifies the data that exists in volatile memory. • `permanent`—Specifies the data that survives reboot. This value is taken as advisory and can be overridden by the explicit device or path in the `cbfDefineFile` object. A given system can support any or all of these values. Max-Access: read-create Status: current	`ephemeral`
cbfDefineFileFormat	`INTEGER { standardBER(1), bulkBinary(2), bulkASCII(3)`	Determines the format of the data in the file: The following are the value definitions: • `standardBER`—Specifies the standard SNMP ASN.1 BER. • `bulkBinary`—Specifies the binary format for this MIB. • `bulkASCII`—Specifies the human-readable form of the `bulkBinary` value. **Note** The details of the formats will be added as comments in this file. A given system can support any or all of these values. Max-Access: read-create Status: current	`bulkBinary`

Table 3-101 File Definition Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cbfDefineFileNow	`INTEGER { notActive(1), ready(2), create(3), running(4)}`	Specifies the control for file creation. The only value that can be set is `create` and that can be set only when the value is `ready`. If the `create` value is set, file creation begins and creates a corresponding entry in `cbfStatusFileTable`. The value is `notActive` as long as the `cbfDefineFileEntryStatus` object or any corresponding entries for the `cbfDefineObjectEntryStatus` object is not active. When `cbfDefineFileEntryStatus` becomes active and all corresponding entries for the `cbfDefineObjectEntryStatuses` object are active, this object automatically goes to `ready`. Max-Access: read-create Status: current	`notActive`
cbfDefineFileEntry Status	`RowStatus`	Allows the control for creation, modification, and deletion of entries. For detailed rules, see the `cbfDefineFileEntry` object in this table. Max-Access: read-create Status: current	none

File Object Table

The object identifiers used for the file object table are listed in Table 3-102.

Table 3-102 File Object Table Object Identifiers

Name	Object Identifier
cbfDefineObjectTable	::= { cbfDefine 10 }
cbfDefineObjectEntry	::= { cbfDefineObjectTable 1 }
cbfDefineObjectIndex	::= { cbfDefineObjectEntry 1 }

The MIB objects used for the file object table are listed in Table 3-103.

Table 3-103 File Object Table MIB Objects

Name	Syntax	Description	Default Value
cbfDefineObjectTable	SEQUENCE OF CbfDefineObjectEntry	Specifies a table of objects to go into the bulk files. Max-Access: not-accessible Status: current	none
cbfDefineObjectEntry	CbfDefineObjectEntry	Provides information about one object for a particular file. An application uses the cbfDefineObjectEntryStatus object to create entries in this table to correspond with entries with the cbfDefineFileTable, which is created first. Entries in this table cannot be changed, created or deleted while the corresponding value of the cbfDefineFileNow object is running. Entries are created by setting the cbfDefineObjectEntryStatus object to either createAndGo or createAndWait. Entries can also be created by setting any or all other read-create columns. For example, the default action is createAndWait. The following are the indexes: • cbfDefineFileIndex • cbfDefineObjectIndex Max-Access: not-accessible Status: current	none
cbfDefineObjectIndex	Unsigned32 (1..4294967295)	Identifies this entry with a unique, arbitrary integer. The numeric order of the entries controls the order of the objects in the file. Max-Access: not-accessible Status: current	none

File Table

The object identifiers used for the file table are listed in Table 3-104.

Table 3-104 File Table Object Identifiers

Name	Object Identifier
cbfStatusFileTable	::= { cbfStatus 5 }
cbfStatusFileEntry	::= { cbfStatusFileTable 1 }
cbfStatusFileIndex	::= { cbfStatusFileEntry 1 }
cbfStatusFileState	::= { cbfStatusFileEntry 2 }

Table 3-104 File Table Object Identifiers (continued)

Name	Object Identifier
cbfStatusFileCompletionTime	::= { cbfStatusFileEntry 3 }
cbfStatusFileEntryStatus	::= { cbfStatusFileEntry 4 }

The MIB objects used for the file table are listed in Table 3-105.

Table 3-105 File Table MIB Objects

Name	Syntax	Description	Default Value
cbfStatusFileTable	`SEQUENCE OF CbfStatusFileEntry`	Provides a table for the bulk file status. Max-Access: not accessible Status: current	none
cbfStatusFileEntry	`CbfStatusFileEntry`	Specifies the status for a particular file. An entry exists in this table each time the `cbfDefineFileNow` object has been set to `create` and the corresponding entry has not been explicitly deleted by the application or bumped to make room for a new entry. Deleting an entry with the `cbfStatusFileState` object running aborts the file creation attempt. The implementation and specific-file system occurs if deleting the entry also deletes the file. Max-Access: not accessible Status: current The following are the indexes: • `cbfDefineFileIndex` • `cbfStatusFileIndex`	none
cbfStatusFileIndex	`Unsigned32 (1..4294967295)`	Identifies this entry with a unique, arbitrary integer. The numeric order of the entries implies how the files were created. Max-Access: not accessible Status: current	none

Table 3-105 File Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cbfStatusFileState	`INTEGER { running(1), ready(2), emptied(3), noSpace(4), badName(5), writeErr(6), noMem(7), buffErr(8), aborted(9)}`	Determines the following file states: • `running`—Specifies that data is being written to the file. • `ready`—Specifies the file is ready to be read. • `emptied`—Specifies an ephemeral file was successfully consumed. • `noSpace`—Specifies no data occurred due to insufficient file space. • `badName`—Specifies no data occurred due to a name or path problem. • `writeErr`—Specifies no data occurred due to a fatal file write error. • `noMem`—Specifies no data occurred due to insufficient dynamic memory. • `buffErr`—Specifies the implementation buffer is too small. • `aborted`—Specifies short is terminated by operator command. Only the `ready` state implies that the file is available for transfer. After the error is implemented, the files are discarded for the specific-file system. Max-Access: read-only Status: current	none
cbfStatusFileCompletion Time	`TimeStamp`	Determines the value of the `sysUpTime`object when the creation attempt completed. A value of 0 indicates not complete. For ephemeral files, this is the time when the `cbfStatusFileState` object goes to `emptied`. For others, this is the time when the state leaves `running`. Max-Access: read-only Status: current	none
cbfStatusFileEntryStatus	`RowStatus`	Specifies the control that allows the entries to be deleted. This object cannot be set to any value other than `destroy`. Max-Access: read-write Status: current	none

Cisco Bulk File Conformance and Compliance Information

The information on conformance listed in Table 3-106 is specific to the Cisco Bulk File.

Table 3-106 Cisco Bulk File Conformance Groups

Name	Object Identifier
ciscoBulkFileMIBConformance	::= { ciscoBulkFileMIB 3 }
ciscoBulkFileMIBCompliances	::={ ciscoBulkFileMIBConformance 1 }
ciscoBulkFileMIBGroups	::={ciscoBulkFileMIBConformance 2 }

Cisco Bulk File Compliance Statement

One object identifier used for compliance is listed in Table 3-107.

Table 3-107 Cisco Bulk File Compliance Object Identifier

Name	Object Identifier
ciscoBulkFileMIBCompliance	::= { ciscoBulkFileMIBCompliances 1 }

One compliance group is listed in Table 3-108.

Table 3-108 Cisco Bulk File MIB Object Used for Compliance

Name	Mandatory Groups	Description	Default Value
ciscoBulkFileMIBCompliance	`ciscoBulkFileDefineGroup`, `ciscoBulkFileStausGroup`	Specifies the compliance statement for entities which implement the Cisco Bulk File MIB. Implementation of this MIB is based on individual product needs. Status: current	none

Cisco Bulk File Units of Conformance

The object identifier for each MIB object are listed in Table 3-109.

Table 3-109 Cisco Bulk File Units of Conformance Object Identifier

Name	Object Identifier
ciscoBulkFileDefineGroup	::= { ciscoBulkFileMIBGroups 1 }
ciscoBulkFileStatusGroup	::= { ciscoBulkFileMIBGroups 2

The MIB objects used for units of conformance are listed in Table 3-110.

Table 3-110 Cisco Bulk File MIB Objects Used for Units of Conformance

Name	Objects	Description	Default Value
ciscoBulkFileDefineGroup	`cbfDefineFileName, cbfDefineFileStorage, cbfDefineFileFormat, cbfDefineFileNow, cbfDefineFileEntryStatus`	Specifies the bulk file definition management. Status: current	none
ciscoBulkFileStatusGroup	`cbfStatusFileState, cbfStatusFileCompletionTime, cbfStatusFileEntryStatus`	Specifies the bulk file status management. Status: current	none

Cisco WAN Statistics Collection Manager

This section describes the MIB objects that reside in the CISCO-WAN-SCM-MIB.my file. The Cisco WAN Statistics Collection Manager MIB module is used to configure the Statistics Collection Manager (SCM) IP addresses, upload statistics files, and download statistics to enable the `stats enable` files in the system.

Note The Cisco WAN Statistics Collection Manager MIB is supported only by PXM45, and defined under the StrataCom Enterprise.

The Cisco WAN SCM MIB objects include:

- Statistics File Information
- Statistics Collection Manager IP Address Configuration Table

Statistics File Information

One object identifier is listed in Table 3-111 for statistics file information.

Table 3-111 Statistics File Information Object Identifier

Name	Object Identifier
cwsStatsFileInfo	::= { cwsFileInfo 1}

One MIB object is listed in Table 3-112 for statistics file information.

Table 3-112 Statistics File Information MIB Object

Name	Syntax	Description	Default Value

cwsStatsFileInfo	`OCTET STRING (SIZE(0..512))`	Contains information on statistics related files.
		
		The statistics files are uploaded and downloaded using a file transfer mechanism FTP, TFTP, and so forth. The downloaded and uploaded files are available from a nonvolatile storage, for example, hard disk, flash disk, and so forth.
		
		The following categories are used for the statistics file:
		
		• `stats upload` file—Contains statistics data. These files can be uploaded only from the switch.
		
		• `stats enable` file—Contains the statistics that have to be enabled or disabled in one or more modules. These files can be downloaded as well as uploaded to or from the switch.
		
		This object can contain one or more records of files that were created, not uploaded in time, or uploaded and downloaded.
		
		Along with the filename, each record contains type of operation (created, uploaded, downloaded major trap, or critical trap) and number of applicable failed attempts.
		
		The values for each of the fields in the OCTET STRING depends upon the file categories `stats upload`, `stats enable`, and so forth.
		
		The following is the object layout:
		
		`NumOfRecords (SIZE(1))`
		`RECORD (SIZE(23))`
		`Filename STRING (SIZE(21))`
		`Reason INT (SIZE(1))`
		`NumOfFailUpload (SIZE(1))`
		
		The following are the definitions for the fields:
		
		• `NumOfRecords`—Designates the number of records. These records can all be the same type and can contain file uploaded information, or `stat enable` information, `stat file` trap information, or `stat enable` trap information.
		
		• `Reason`—Specifies the following fields:
		
		– 1—Indicates the file is created or uploaded.
		
		– 2—Indicates a MAJOR warning for the file not being uploaded.
		
		– 3—Indicates a CRITICAL warning for the file not being uploaded.
			none

Table 3-112 Statistics File Information MIB Object (continued)

Name	Syntax	Description	Default Value
cwsStatsFileInfo (continued)		– 4—Indicates that the SNMP manager has downloaded the `stats enable` file. – 5—Indicates that the module, for example, Processor Module or Service Module, accepted the `stats enable` file without any error. The modules store the statistics information to enable or disable a dependent mechanism. This value cannot be set from the SNMP Manager. – 6—Indicates that the module did not accept the `stats enable` file due to some error. This value cannot be set from the SNMP manager. • `NumOfFailUpload`—Speechifies the value is set to 0 except or for the cases when the `Reason` field is either 1 (created), 2, 3 (`stat file` trap). The parameter can have a value greater than 0. Max-Access: read-write Status: current	

Statistics Collection Manager IP Address Configuration Table

The object identifiers for the SCM IP Address Configuration Table are listed in Table 3-113.

Table 3-113 SCM IP Address Configuration Table Object Identifiers

Name	Object Identifier
cwsConfTable	::= { cwsConfAddress 1 }
cwsConfEntry	::= { cwsConfTable 1 }
cwsConfIndex	::= { cwsConfEntry 1}
cwsConfIpAddress	::= { cwsConfEntry 2 }
cwsConfRowStatus	::= { cwsConfEntry 3 }

The MIB objects for the SCM IP Address Configuration Table are listed in Table 3-114.

Table 3-114 SCM IP Address Configuration Table MIB Objects

Name	Syntax	Description	Default Value
cwsConfTable	`SEQUENCE OF CwsConfEntry`	Contains the entries for the IP Addresses of SCM. Max-Access: not-accessible Status: current	none
cwsConfEntry	`CwsConfEntry`	Provides an entry for the SCM IP Address. Max-Access: not-accessible Status: current The index is `cwsConfIndex`.	none
cwsConfIndex	`Unsigned32 (1..32)`	Specifies the unique value for the entry in the table. The following are the settings for the IP address: • Entry 1 is called the primary IP address. • Entry 2 is called the secondary IP address. • Entry 3 is called the tertiary IP address. Max-Access: not-accessible Status: current	none
cwsConfIpAddress	`IpAddress`	Determines the IP Address for SCM. Max-Access: read-create Status: current	none
cwsConfRowStatus	`RowStatus`	Adds, deletes, and modifies the IP address configuration of the SCM. The row is created by setting this object to `createAndGo(4)`. A row is deleted by setting this object to `destroy(6)`. The `cwsConfRowStatus` object cannot be set to a value other than `createAndGo(4)` and `destroy(6)`. Max-Access: read-create Status: current	none

Cisco WAN Statistics Collection Manager Conformance and Compliance Information

The information on conformance is specific to the Cisco WAN Statistics Collection Manager in Table 3-115.

Table 3-115 Statistics Collection Manager Conformance Groups

Name	Object Identifier
cwsMIBConformance	::= {ciscoWanScmMIB 3}
cwsMIBCompliances	::= {cwsMIBConformance 1}
cwsMIBGroups	::= {cwsMIBConformance 2}

Cisco WAN Statistics Collection Manager Compliance Statement

One object identifier used for compliance is listed in Table 3-116.

Table 3-116 Statistics Collection Manager Compliance Object Identifier

Name	Object Identifier
cwsMIBCompliance	::= {cwsMIBCompliances 1}

One compliance group is listed in Table 3-117.

Table 3-117 Statistics Collection Manager MIB Object Used for Compliance

Name	Mandatory Groups	Description	Default Value
cwsMIBCompliance	`cwsFileGroup`, `cwsConfGroup`	Specifies the compliance statement for the SCM IP Address configuration group. Status: current	none

Cisco WAN Statistics Collection Manager Units of Conformance

The object identifier for each MIB object are listed in Table 3-118.

Table 3-118 Statistics Collection Manager Units of Conformance Object Identifiers

Name	Object Identifier
cwsFileGroup	::= { cwsMIBGroups 1}
cwsConfGroup	::= { cwsMIBGroups 2}

The MIB objects used for units of conformance are listed in Table 3-119.

Table 3-119 Statistics Collection Manager MIB Objects Used for Units of Conformance

Name	Objects	Description	Default Value
cwsFileGroup	`cwsStatsFileInfo`	Specifies the objects related to the statistics file name. Status: current	none
cwsConfGroup	`cwsConfIpAddress`, `cwsConfRowStatus`	Specifies the objects related to configuring the SCM IP addresses. Status: current	none

Cisco System

This section describes the MIB objects used for a standard set of basic system information. The Cisco System MIB resides in the CISCO-SYSTEM-MIB.my file.

Note The Cisco System MIB is supported only by PXM45.

The Cisco System MIB objects include:

- Clock Section
- Location Section
- Cisco System Conformance Information

Clock Section

The object identifiers for the clock section are listed in Table 3-120.

Table 3-120 Cisco System Clock Section Object Identifiers

Name	Object Identifier
csyClockDateAndTime	::= { csyClock 1 }
csyClockLostOnReboot	::= { csyClock 2 }

The clock section MIB objects are listed in Table 3-121.

Table 3-121 Cisco System Clock Section MIB Objects

Name	Syntax	Description	Default Value
csyClockDateAndTime	`DateAndTime`	Determines the current local date and time for the system. The settings for this object is equivalent to the settings of an automated clock and calendar. The value of the object tracks the date and time from the value set. **Note** Due to hardware limitations, some systems may not be able to preserve such meaning across reboots of the system, for example, `csyClockLostOnReboot`. A constant value of all zeros and length 8 indicates the system is not aware of the present date and time. This object can be read-only on some systems. Max-Access: read-write Status: current	none
csyClockLostOnReboot	`TruthValue`	Indicates whether the system can preserve knowledge of the current date and time across a system reboot. A value of `true` indicates the clock must be reset from some external source each time the system reboots. A value of `false` indicates the system has the ability to keep time across reboots. Max-Access: read-only Status: current	none

Location Section

One object identifier is listed in Table 3-122 for the location section.

Table 3-122 Cisco System Location Section Object Identifier

Name	Object Identifier
csyLocationCountry	::= { csyLocation 1 }

One MIB object is listed in Table 3-123 for the location section.

Table 3-123 Cisco System Location Section MIB Object

Name	Syntax	Description	Default Value
csyLocationCountry	CountryCode	Determines the country where the system is physically located. On some systems and for some technologies, this value affects behavior, such as standards for communication. All technologies can default by using the setting of this value, but can provide an override if necessary. The default value of this object is us. Systems destined for other countries can use a different default. Systems which the value does not affect operation can default to a zero-length value. Max-Access: read-write Status: current	none

Cisco System Conformance Information

The object identifiers are listed in Table 3-124 for conformance.

Table 3-124 Cisco System Conformance Groups

Name	Object Identifier
ciscoSystemMIBConformance	::= { ciscoSystemMIB 3 }
ciscoSystemMIBCompliances	::={ ciscoSystemMIBConformance 1 }
ciscoSystemMIBGroups	::={ ciscoSystemMIBConformance 2 }

Cisco System Compliance Statement

The compliance object identifier is listed in Table 3-125.

Table 3-125 Cisco System Compliance Object Identifier

Name	Object Identifier
csyClockDateAndTime	::= { ciscoSystemMIBCompliances 1 }

The objects for compliance are listed in Table 3-126.

Table 3-126 Cisco System Compliance Objects

Name	Groups	Description	Default Value
ciscoSystemMIBCompliance	ciscoSystemClockGroup, ciscoSystemLocationGroup	Implements the Cisco System MIB for the entities of the compliance statement. Adherence to this compliance statement is expected of all Cisco systems. Status: current	none
csyClockDateAndTime	none	Specifies that write access is not implemented on systems that do not allow their clock to be set either because they have a different, more reliable source or they do not use the same information. Min-Access: read-only	none

Cisco System Units of Conformance

The object identifiers for Cisco System units of conformance are listed in Table 3-127.

Table 3-127 Cisco System Units of Conformance Object Identifiers

Name	Object Identifier
ciscoSystemClockGroup	::= { ciscoSystemMIBGroups 1 }
ciscoSystemLocationGroup	::= { ciscoSystemMIBGroups 2 }

The objects are listed in Table 3-128 for the Cisco System units of conformance.

Table 3-128 Cisco System Objects Used for Units of Conformance

Name	Objects	Description	Default Value
ciscoSystemClockGroup	csyClockDateAndTime, csyClockLostOnReboot	Determines the clock attributes. Status: current	none
ciscoSystemLocationGroup	csyLocationCountry	Determines the physical location attributes. Status: current	none

ATM Connection Statistics

This section describes the MIB objects that are used for ATM switch connection related real-time statistical counter objects. The ATM Connection Statistics MIB resides in the CISCO-WAN-ATM-CONN-STAT-MIB.my file.

Note The ATM Connection Statistics MIB is supported only by AXSM-E and defined under the StrataCom Enterprise.

The object identifiers for the ATM Connection Statistics MIB are listed in Table 3-129.

Table 3-129 ATM Connection Statistics Object Identifiers

Name	Object Identifier
cwacsIngRcvCLP0	::= { cwacsEntry 1 }
cwacsIngRcvCLP1	::= { cwacsEntry 2 }
cwacsIngCLP0UpcDiscard	:= { cwacsEntry 7 }
cwacsIngCLP1UpcDiscard	::= { cwacsEntry 8 }
cwacsIngCLP0UpcTagged	::= { cwacsEntry 9 }
cwacsIngRcvEFCI1	::= { cwacsEntry 11 }
cwacsIngRcvEOF1	::= { cwacsEntry 20 }
cwacsIngVCQueueDepth	::= { cwacsEntry 23 }
cwacsEgrRcvCLP0	::= { cwacsEntry 24 }
cwacsEgrRcvCLP1	::= { cwacsEntry 25 }
cwacsEgrRcvEFCI1	::= { cwacsEntry 31 }
cwacsEgrRcvEOF1	::= { cwacsEntry 40 }
cwacsEgrVCQueueDepth	::= { cwacsEntry 43 }
cwacsStatsClear	::= { cwacsEntry 44 }

The ATM Connection Statistics MIB objects are listed in Table 3-130.

Table 3-130 ATM Connection Statistics MIB Objects

Name	Syntax	Description	Default Value
cwacsIngRcvCLP0	Counter32	Determines the number of valid CLP=0 ATM cells in the ingress direction of the channel prior to the traffic management entity. Units: cells Max-Access: read-only Status: current	none
cwacsIngRcvCLP1	Counter32	Determines the number of valid CLP=1 ATM cells in the ingress direction of the channel prior to the traffic management entity. Units: cells Max-Access: read-only Status: current	none
cwacsIngCLP0Upc Discard	Counter32	Determines the number of valid CLP=0 cells in the ingress direction of the channel that are discarded at the traffic policing entity due to Usage Parameter Control (UPC) or Peak Cell Rate (PCR) parameter violation. If incoming cells violate established user-network contract, the UPC operation checks the validity of the user traffic on a connection. If the cell violates the PCR parameter for the connection, it is discarded or the cell is marked low priority by setting CLP=1. Units: cells Max-Access: read-only Status: current	none
cwacsIngCLP1Upc Discard	Counter32	Determines the number of valid CLP=1 cells in the ingress direction of the channel that are discarded at the traffic policing entity due to the UPC or PCR parameter violation. Units: cells Max-Access: read-only Status: current	none
cwacsIngCLP0Upc Tagged	Counter32	Determines the number of valid CLP=0 cells in the ingress direction of the channel that are tagged as low priority by the UPC. Units: cells Max-Access: read-only Status: current	none
cwacsIngRcvEFCI1	Counter32	Determines the number of valid EFCI=1 cells in the ingress direction of the channel before the class of service queue. Units: cells Max-Access: read-only Status: current	none

Table 3-130 ATM Connection Statistics MIB Objects (continued)

Name	Syntax	Description	Default Value
cwacsIngRcvEOF1	Counter32	Determines the number of valid EOF=1 ATM cells in the ingress direction of the channel prior to the traffic management entity. Units: cells Max-Access: read-only Status: current	none
cwacsIngVCQueue Depth	Gauge32	Determines the current length of the ingress queue of the channel in terms of the number of cells. Units: cells Max-Access: read-only Status: current	none
cwacsEgrRcvCLP0	Counter32	Determines the number of valid CLP=0 ATM cells in the egress direction of the channel prior to the traffic management entity. Units: cells Max-Access: read-only Status: current	none
cwacsEgrRcvCLP1	Counter32	Determines the number of valid CLP=1 ATM cells in the egress direction of the channel prior to the traffic management entity. Units: cells Max-Access: read-only Status: current	none
cwacsEgrRcvEFCI1	Counter32	Determines the number of valid EFCI=1 cells in the egress direction of the channel before the class of service queue. Units: cells Max-Access: read-only Status: current	none
cwacsEgrRcvEOF1	Counter32	Determines the number of valid EOF=1 ATM cells in the egress direction of the channel prior to the traffic management entity. Units: cells Max-Access: read-only Status: current	none

Table 3-130 ATM Connection Statistics MIB Objects (continued)

Name	Syntax	Description	Default Value
cwacsEgrVCQueue Depth	`Gauge32`	Specifies the current length of the egress queue of the channel in terms of number of cells. Units: cells Max-Access: read-only Status: current	none
cwacsStatsClear	`TruthValue`	Specifies that the agent clears the channel statistics collection if it receives a value of `true(1)` request from a manager. Setting a value of `false(2)` has no implication. Max-Access: read-write Status: current	`false`

ATM Connection Statistics Compliance Information

The object identifiers that are used for compliance are listed in Table 3-131.

Table 3-131 ATM Connection Statistics Compliance Object Identifiers

Name	Object Identifier
ciscoWanAtmConnStatMIBConformance	::= { ciscoWanAtmConnStatMIB 3 }
ciscoWanAtmConnStatMIBCompliances	::= { ciscoWanAtmConnStatMIBConformance 1 }
ciscoWanAtmConnStatMIBGroups	::= { ciscoWanAtmConnStatMIBConformance 2 }

ATM Connection Statistics Conformance Information

One object identifier is listed in Table 3-132 for conformance.

Table 3-132 ATM Connection Statistics Conformance Object Identifier

Name	Object Identifier
ciscoWanAtmConnStatMIBCompliance	::= { ciscoWanAtmConnStatMIBCompliances 1 }

One object is listed in Table 3-133 for conformance.

Table 3-133 ATM Connection Statistics Object Used for Conformance

Name	Object	Description	Default Value
ciscoWanAtmConnStatMIBCompliance	`cwacsGroup`	Describes the compliance statement for the Cisco statistics MIB. Status: current	none

ATM Connection Statistics Units of Conformance

One object identifier is listed in Table 3-134 for the ATM Connection Statistics units of conformance.

Table 3-134 ATM Connection Statistics Units of Conformance Object Identifier

Name	Object Identifier
cwacsGroup	::= { ciscoWanAtmConnStatMIBGroups 1 }

One object is listed in Table 3-135 for the ATM Connection Statistics units of conformance.

Table 3-135 ATM Connection Statistics Object Used for Units of Conformance

Name	Objects	Description	Default Value
cwacsGroup	cwacsIngRcvCLP0, cwacsIngRcvCLP1, cwacsIngXmtCLP0, cwacsIngXmtCLP1, cwacsIngCLP0CoSDiscard, cwacsIngCLP1CoSDiscard, cwacsIngCLP0UpcDiscard, cwacsIngCLP1UpcDiscard, cwacsIngCLP0UpcTagged, cwacsIngRcvEFCI0, cwacsIngRcvEFCI1, cwacsIngEFCI0Discard, cwacsIngEFCI1Discard, cwacsIngRcvOAM, cwacsIngOAMDiscard, cwacsIngRcvRM, cwacsIngRMDiscard, cwacsIngXmtFRm, cwacsIngXmtBRmFsRm, cwacsIngRcvEOF1, cwacsIngEOF1Discard, cwacsIngACR, cwacsIngVCQueueDepth, -- Egress statistics cwacsEgrRcvCLP0, cwacsEgrRcvCLP1, cwacsEgrXmtCLP0, cwacsEgrXmtCLP1, cwacsEgrCLP0CoSDiscard, cwacsEgrCLP1CoSDiscard, cwacsEgrRcvEFCI0, cwacsEgrRcvEFCI1, cwacsEgrEFCI0Discard, cwacsEgrEFCI1Discard, cwacsEgrRcvOAM, cwacsEgrOAMDiscard, cwacsEgrRcvRM, cwacsEgrRMDiscard, cwacsEgrXmtFRm, cwacsEgrXmtBRmFsRm, cwacsEgrRcvEOF1, cwacsEgrEOF1Discard, cwacsEgrACR, cwacsEgrVCQueueDepth, cwacsStatsClear	Provides traffic statistic information that are related to an ATM Channel. Status: current	none

Robust Trap Mechanism MIB Objects

This section describes the MIB objects that make up Robust Trap Mechanism (RTM). The RTM MIBs reside in the RTM-MIB.my file.

Note The RTM MIB is supported by PXM45 and defined under the StrataCom Enterprise.

The RTM objects include:

- trapConfigTable Group
- trapUploadTable Group

trapConfigTable Group

The object identifiers used for the trapConfigTable are listed in Table 3-136.

Table 3-136 trapConfigTable Object Identifiers

Name	Object Identifier
trapConfigTable	::= { trapsConfig 1 }
trapConfigEntry	::= { trapConfigTable 1 }
managerIPaddress	::= { trapConfigEntry 1 }
managerPortNumber	::= { trapConfigEntry 2}
managerRowStatus	::= { trapConfigEntry 3}
readingTrapsFlag	::= { trapConfigEntry 4}
nextTrapSeqNum	::= { trapConfigEntry 5}
managerNumOfValidEntries	::= { trapsConfig 2 }
lastSequenceNumber	::= { trapsConfig 3}

The MIB objects are listed in Table 3-137.

Table 3-137 trapConfigTable Group MIB Objects

Name	Syntax	Description	Default Value
trapConfigTable	`SEQUENCE OF TrapConfigEntry`	Provides the table that has information on the manager receiving traps. Access: not-accessible Status: mandatory	none
trapConfigEntry	`TrapConfigEntry`	Determines an entry for the manager. Access: not-accessible Status: mandatory The index is `managerIPaddress`.	none

Table 3-137 trapConfigTable Group MIB Objects (continued)

Name	Syntax	Description	Default Value
managerIPaddress	`IpAddress`	Determines the IP address that the manager used as an index to the table. This manager receives the traps from AXIS. Access: read-write Status: mandatory	none
managerPortNumber	`INTEGER`	Determines the port number that the manager receives traps from an agent. Access: read-write Status: mandatory	none
managerRowStatus	`INTEGER { addRow (1), delRow (2)}`	Indicates when the manager registers with the agent. If applicable, this object is set to `addRow`, and is checked by the manager keep-alive request. Access: read-write Status: mandatory	none
readingTrapsFlag	`INTEGER { false(1), true(2)}`	Determines an entry for the state of the manager either reading or not reading traps from the agent. Access: read-write Status: mandatory	none
nextTrapSeqNum	`INTEGER`	Indicates the first trap that the manager is interested in uploading The agent updates the `nextTrapPointer` object internally to point to FIFO trap with sequence number equal to `nextTrapSeqNum`. If there is no trap in the FIFO with this sequence number, the agent sets the `nextTrapPointer` object to the head of FIFO and returns an error response. Access: read-write Status: mandatory	none
managerNumOfValid Entries	`INTEGER (0..8)`	Determines the number of managers in the table that are programmed to receive traps. Access: read-only Status: mandatory	none
lastSequenceNumber	`INTEGER`	Determines the sequence number of the last trap generated on the agent. Access: read-only Status: mandatory	none

trapUploadTable Group

The object identifiers used for the trapConfigTable are listed in Table 3-138.

Table 3-138 trapUploadTable Group Object Identifiers

Name	Object Identifier
trapUploadTable	::= { trapsConfig 4}
trapUploadEntry	::= { trapUploadTable 1 }
trapManagerIPaddress	::= { trapUploadEntry 1 }
trapSequenceNum	::= { trapUploadEntry 2}
trapPduString	::= { trapUploadEntry 3}
endOfQueueFlag	::= { trapUploadEntry 4}

The MIB objects are listed in Table 3-139.

Table 3-139 trapUploadTable Group MIB Objects

Name	Syntax	Description	Default Value
trapUploadTable	`SEQUENCE OF TrapUploadEntry`	Specifies the table, which is used by the manager, retrieves missing traps by using the robust trap mechanism. The manager does a `GetRequest` operation on this table. Access: not-accessible Status: mandatory	none
trapUploadEntry	`TrapUploadEntry`	Indicates the manager does a `GetRequest` operation on the elements of this entry to upload missing traps. Access: not-accessible Status: mandatory The index is `trapManagerIPaddress`.	none
trapManagerIPaddress	`IpAddress`	Specifies the IP address the manager used as an index to the table. This manager receives the traps from AXIS. Access: read-write Status: mandatory	none
trapSequenceNum	`INTEGER`	Determines the sequence number associated with the trap. Access: read-only Status: mandatory	none

Table 3-139 trapUploadTable Group MIB Objects (continued)

Name	Syntax	Description	Default Value
trapPduString	`DisplayString`	Specifies the trap description string. Access: read-only Status: mandatory	none
endOfQueueFlag	`INTEGER { false (1), true (2)}`	Indicates the flag that is saved in the agent FIFO queue for the last trap. Access: read-only Status: mandatory	none

Error Status MIB Objects

This section describes the MIB objects that reside in the ERR-STATUS-MIB.my file. The Error Status table is the only MIB defined in the Error Status MIB group.

Note The Error Status table is supported by PXM45 and defined under the StrataCom Enterprise.

Error Status Table

The Error Status table maintains status information about SNMP requests from a manager. When an SNMP request is completed, an entry is created. In addition to a successful status, the manager provides information associated with any failed requests. This table is indexed by the request ID. The request ID is the SNMP PDU identifier associated with the failed request.

Currently a limit of 100 entries are imposed on this table. Each manager is allowed to access their own table. Individual manager access is determined by the IP address.

The object identifiers used for the Error Status Table are listed in Table 3-140.

Table 3-140 Error Status Table Object Identifiers

Name	Object Identifier
strmErrors	::= { stratacom 910 }
errStatusLastIndex	::= { strmErrors 1 }
errStatusTable	::= { strmErrors 2 }
errStatusTableEntry	::= { errStatusTable 1 }
errReqId	::= { errStatusTableEntry 1 }
errCode	::= { errStatusTableEntry 2 }
errStatusDesc	::= { errStatusTableEntry 3 }

The MIB objects are listed in Table 3-141.

Table 3-141 Error Status Table MIB Objects

Name	Syntax	Description	Default Value
strmErrors	none	Contains the value of the request ID for the current entry in the error status table. Many management applications have no knowledge of the request IDs used by the SNMP engine. The management applications can get the value of this object, and then use it to retrieve the error status for the last `Set` operation.	none
errStatus LastIndex	`INTEGER`	If at least one entry exists in the error status table for the given manager, the value of this object contains the index corresponding to the last entry. If no entry exists in the error status table for the given manager, the value of this object is -1. Access: read-only Status: mandatory	none
errStatus Table	`SEQUENCE OF ErrStatusTableEntry`	Provides a list of error statuses for a given SNMP manager. Access: not-accessible Status: mandatory	none
errStatus TablEntry	`ErrStatusTableEntry`	Provides a general error status entry. Access: not-accessible Status: mandatory The index is `errReqId`.	none
errReqId	`INTEGER`	Contains the PDU request ID associated with the error. Access: read-only Status: mandatory	none

Table 3-141 Error Status Table MIB Objects (continued)

Name	Syntax	Description	Default Value
errCode	INTEGER { success(1), existErr(2), syntaxErr(3),resourceErr(4), databaseLocked(5),otherErr(6), wrongType(7), wrongLength(8), wrongEncoding(9),wrongValue(10), noCreation(11), inconsistentValue(12), resourceUnavailable(13), commitFailed(14),undoFailed(15), authorizationError(16), notWritable(17),inconsistentName(18), featureDisabled(19), m32Problem(20),sarProblem(21), bnmProblem(22), ascUpdFailed(23), lineEnabled(24),lineDisabled(25), lmMismatch(26),lineHasPorts(27), portEnabled(28),portDisable(29), portHasChan(30),chanEnabled(31), chanDisabled(32),dlciEnabled(33), dlciDisabled(34),ovrsubconnPass(35), ovrsubConnFail(36),portVpiVciInUse(37), invalidPrimarySlot(38), invalidSecondarySlot(39),linkFull(40), primaryDuplicate(41), secondaryDuplicate(42), primaryNotPresent(43), secondaryNotPresent(44), srmNotPresent(45),invalidCommand(46), invalidCardType(47),featureMismatch(48), lmiEnabled(49),dlciUsed(50), invalidRedType(51), bertResourcesNotFree(52), bertResourcesNotReady(53),	Contains an error status code that is used by the manager to take automated corrective actions when the requests encounter failures. If no error is encountered, a successful status is provided. Access: read-only Status: mandatory	none

Table 3-141 Error Status Table MIB Objects (continued)

Name	Syntax	Description	Default Value
errCode (continued)	bertSlotEmpty(54), bertUnsupportedCard(55), bertNotOwner(56),bertStartFailed(57), bertModFailed(58),bertDelFailed(59), bertUnsupportedType(60), bertWrongParams(61),bertUnableToFree(62), bertGeneralError(63),portInLoopback(64), invalidT3LineNum(65),invalidT1LineNum(66), invalidSlotNum(67),invalidLineNum(68), notEnoughLine(69), lineInUse(70),t3NotEnabled(71), smNotPresent(72),smNotPrimary(73), srm3t3NotPresent(74),lineInLoopback(75), lineInconsistentLoopback(76), lineLoopNotAllowed(77), versionMismatch(78),portOutOfService(79), lineOutOfService(80), bertNotConfigured(81), bertConfigurationIncomplete(82), testAlreadyOn(83),testNotOn(84), loopUpFailure(85),loopDownFailure(86), bertPatternSyncFailure(87), -- error codes added for PAR 100 - 199 localIfNotOk(100),remoteIfNotOk(101), localChannelsNotEnough(102), remoteChannelsNotEnough(103), localAddrNotOk(104),remoteAddrNotOk(105), localAddrExist(106),remoteAddrExist(107), invalidConnAddr(108), maxConnsExceeded(109), lmiTrkNotAdded(110),connNotExist(111), vpiConflictForVcc(112), vpiConflictForVpc(113),vpiRangeErr(114), vciRangeErr(115),invalidConnType(116), genConnErr(117), masterConnAllocFailed(118), slaveConnAllocFailed(119), masterLogepAllocFailed(120), slaveLogepAllocFailed(121), masterCmtFailed(122),		

Table 3-141 Error Status Table MIB Objects (continued)

Name	Syntax	Description	Default Value
errCode (continued)	slaveCmtFailed(123), daxCmtFailed(124), masterChgFailed(125), laveChgFailed(126),daxChgFailed(127), masterDelFailed(128),slaveDelFailed(129), daxDelFailed(130),masterTimeout(131), mxConAddSessionExceeded(132), masterSessionUnavail(133), slaveSessionUnavail(134), unknownDest(135),localAddrUnknown(136), rmtAddrUnknown(137), portHasResPart(138),networkBusy(139), slaveCnfconNotAllowed(140), -- error codes added for SVC-pnni 200 - 249 pnniConfigurationFail(200), pnniInvalidValue(201), pnniTargtNodeNotExist(202), pnniEntryExist(203), pnniEntryNotExist(204), pnniTargetNodeAdminUp(205), pnniTargetInterfaceNotExist(206), pnniInvalidAtmEndStationAddress(207), pnniNotReadyForSet(208), -- error codes aded for SVC-ccb 250 - 349 ccbNullMessageSent(250), ccbOutOfMemory(251), ccbSSIMessageAllocationFailed(252), ccbSwitchResponseReturnedFailure(253), ccbOperationNotPermittedOnControlPort(254), ccbPortExists(255),ccbPortNotExist(256), ccbPortInUseBySwitch(257), ccbPortNotOutOfService(258), ccbPortUpAlready(259), ccbPortDownAlready(260), ccbPortTypeNotMatch(261), ccbAtmAddrExceedMax(262), ccbAtmAddrNotExist(263), ccbAtmAddrExist(264), ccbAtmAddrNotAllowed(265), ccbAtmAddrSendPnniFailed(266), -- error code added for SPVC 350 - 449 vpiVciNotAvail(350),rmtAddrRequired(351), endptAllocFailed(352),legAllocFailed(353), invalidTrafficParam(354),daxNoSlave(355),		

Table 3-141 Error Status Table MIB Objects (continued)

Name	Syntax	Description	Default Value
errCode (continued)	daxSlaveNotAvail(356), endptNotExist(357), endptNotPersistent(358), delFailed(359),modFailed(360), upFailed(361),downFailed(362), rerouteFailed(363),operNotAppl(364), connAlreadyUp(365),connAlreadyDown(366), daxSlaveParamMismatch(367), daxSlaveTrafficParamModNotAllowed(368), daxRerouteNotAppl(369),interfaceDown(370), spvcNotAllowedOnPart(371), spvcBlockEnabled(372), spvcNotAllowedOnIntf(373), spvcNotAllowedDelDaxSlaveEp(374), noPepToDelete(375),portDontSuppoSc(376), lscrBiggerThanLpcr(377), rscrBiggerThanRpcr(378), cbrNeedsLpcr(379),cbrNeedsRpcr(380), vbrNeedsLpcr(381),vbrNeedsRpcr(382), abrNeedsLpcr(383),abrNeedsRpcr(384), badRcdv(385),badRctd(386), badRMaxClr(387),lCrTooBig(388), rCrTooBig(389), ctdNotApplicableServiceType(390), cdvNotApplicableServiceType(391), vccExistsVpi(392),epAlreadyExists(393), blockedInNoProv(394), commandProcNotReady(601), invalidIfIndex(602),invalidShelfNum(603), invalidBayNum(605),invalidPortNum(606), provDisBcMismatch(607), configNotSupported(608), frontcardMismatch(609), portRateSumExceedLnRate(610), axsmDiskErr(611), portRateExceedLnRate(612), cannotModSctID(613),partNotExist(614), partExist(615),badBayOrLine(616), vsiSetErr(617),vsisAddPart(618), vsisCnfPart(619),vsisDelPart(620), partRateExceedsPortRate(621), lineResvForE3(622),lineResvForT3(623), minRateExceedMax(624),vpiLoExceedHi(625),		

Table 3-141 Error Status Table MIB Objects (continued)

Name	Syntax	Description	Default Value
errCode (continued)	vciLoExceedHi(626),swErr(627), actsctCntExceed(628),sctFileMissing(629), sctFileBad(630),sctSgParmMismatch(631), hardwareConfigErr(633), vpiOverlap(634),vciOverlap(635), cannotAllocLcn(636),cannotAllocBw(637), partPortParmMismatch(638), cannotReleaseLcn(639),cannotReleaseBw(640), cannotModLcnRange(641), cannotModBwRange(642),invalidPartNum(643), invalidCtlrNum(644), invalidSlave(645),invalidConnref(646), dupRequest(647),pendingRequest(648), outOfMemory(649),resourceInUse(650), dcMismatch(651),dcMissing(652), bcMismatch(653),bcMissing(654), invalidDiskRecord(655), invalidCellMapping(656), invalidLineTrace(657), invalidTcaSeverity(658),invalidAtmPhy(659), conflictingPort(660), minConnsExceedsMax(661), partTblFull(662),frozenStatsLvl(663), invalidStatsLvl(664),invalidImaGroup(665), imaGroupEnable(666),imaGroupDisable(667), imaGroupHasPorts(668), cannotCnfProtLine(669), workingLineDown(670), -- error code added for RPM 701 - 800 invalidParam(701),badIpcData(702), rvtNotActive(703),diskUpdFailed(704), subIfDown(705), vpiOutOfRange(706), vciOutOfRange(707),vcdOutOfRange(708), inarpNotAllowed(709),vcdInUse(710), secondVccOnPTP(711),unknownLANEType(712), secondIlmiVc(713),unknownEncapType(714), pppErr(715),invalidVpi(716), invalidVci(717), serviceTypeNotSupported(718), cantChgVpcFlag(719),cantChgRmtVpi(720), cantChgRmtVci(721),cantChgRmtAddr(722), cantChgMastership(723),cantChgSubIf(724), cantChgServType(725),cantCnfVc(726), cantDelVc(727),connNotFound(728),		

Table 3-141 Error Status Table MIB Objects (continued)

Name	Syntax	Description	Default Value
errCode (continued)	subIfNotExist(729),endptNotMaster(730), ctlrIdRequired(731), ingBwRangeRequired(732), egrBwRangeRequired(733), vpiRangeRequired(734),vciRangeRequired(735), invalidIngBwRange(736), invalidEgrBwRange(737), ingBwNotAvail(738),egrBwNotAvail(739), invalidVpiRange(740),invalidVciRange(741), vpiRangeInUse(742),vciRangeInUse(743), cantChgCtlrId(744),cantChgPartType(745), cantModPart(746),cantDelPart(747), ubIfExist(748),invalidIpAddr(749), subIfTypeRequired(750), cantChgSubIfType(751), ssubIfNotFound(752), cantShutMainIf(753),cantDelMainIf(754), rpmNotActive(755),ipcErr(756), rpmNoResponse(757)}		
errStatusDesc	DisplayString	Contains error status information for the failed Set types to one or more objects. This object is updated only on failed Set types. Since the SNMP standard allows only a limited number of error status returns, the managers can retrieve this variable to get additional information on a failed Set operation. Typically this object can contain information specifying date, Set or Get operation, current SNMP table, and additional error information. Access: read-only Status: mandatory	none

Basis Shelf MIB Objects

This section describes the individual MIB objects that make up the PXM45 MIB-specific files. The BASIS SHELF MIB file.

Note The BASIS SHELF MIB is supported by PXM45 and defined under the StrataCom Enterprise.

The Basis Shelf MIB objects include:

- Shelf Table
- axisRedundancy Group

Shelf Table

The object identifier for each MIB object is listed in Table 3-142.

Table 3-142 Shelf Table Object Identifiers

Name	Object Identifier
shelfTable	::= { basisShelf 1 }
shelfEntry	::= { shelfTable 1}
shelfNum	::= { shelfEntry 1 }
shelfSlotNum	::= { shelfEntry 2 }
shelfBkplnSerialNumDeprecated	::= { shelfEntry 3 }
shelfFunctionModuleState	::= { shelfEntry 4 }
shelfFunctionModuleType	::= { shelfEntry 5 }
shelfFunctionModuleHoldReset	::= { shelfEntry 6 }
statsMasterIpAddress	::= { basisShelf 10 }
statsCollectionInterval	::= { basisShelf 11 }
statsBucketInterval	::= { basisShelf 12 }
userName	::= { basisShelf 13 }
shelfIntegratedAlarm	::= { basisShelf 14 }

The MIB objects are listed in Table 3-143.

Table 3-143 Shelf Table MIB Objects

Name	Syntax	Description	Default Value
shelfTable	`Sequence of shelfEntry`	Provides the physical slot number and state of the cards for this table. Access: not-accessible Status: mandatory	none
shelfEntry	`shelfEntry`	Provides an entry for the slot. Access: not-accessible Status: mandatory The indexes are the following: • `shelfNum` • `shelfSlotNum`	none
shelfNum	`Integer (1..4)`	Represents a unique value for each shelf. The value ranges from one to the maximum value of `shelfNumber`. Access: read-only Status: mandatory	none

Table 3-143 Shelf Table MIB Objects (continued)

Name	Syntax	Description	Default Value
shelfSlotNum	`Integer (1..33)`	Displays the slot number of a card. BASIS can have a maximum of 16 slots in one shelf. A value of 17 is used to reset the whole shelf. The Cisco MGX 8850 can have a maximum of 32 slots in one shelf. A value of 33 is used to reset the whole shelf. Access: read-only Status: mandatory	none
shelfBkplnSerialNum Deprecated	`Integer (1..4)`	Specifies that this object is deprecated. It is available as the ninth entry in `basisShelf`. It is included here to prevent a hole in the `shelfEntry` table. Access: read-only Status: mandatory	none
shelfFunctionModule State	`Integer {nocard (1), standby (2), active (3), failed (4), selfTest (5), heldInReset (6), boot (7), mismatch (8), unknown (9), coreCardMismatch (10), blocked (11), reserved (12), (hold state applies to PXM during Graceful Upgrade) hold (13)}`	Holds the status of a card in a particular shelf-slot. The Slave PXM assumes the hold state during PXM upgrades. In this state, the Slave PXM is running a different firmware but is receiving all standby updates from Battery-backed nonvolatile RAM (BRAM) and the Database. Access: read-only Status: mandatory	none
shelfFunctionModule Type	`Integer {other (1), asc (2), Reserved for MGX 8850 after 1000: pxm1 (1000), pxm1-2t3e3 (1001), pxm1-4oc3 (1002), pxm1-oc12 (1003), rpm (2000)}`	Indicates the type of a card in a particular shelf-slot. Access: read-only Status: mandatory	`other`
shelfFunctionModule HoldReset	`Integer {doNotHold(1), holdInReset(2)}`	Retains the card in a particular shelf-slot during reset if this object is set to `holdInReset`. If the object is set to `doNotHold`, it resets only the card in the specified slot. If the slot number is 17, it resets the whole shelf. Access: read-only Status: mandatory	none
statsMasterIpAddress	`IpAddress`	Displays the IP address of the `statsMaster`. Access: read-only Status: mandatory	none

Table 3-143 Shelf Table MIB Objects (continued)

Name	Syntax	Description	Default Value
statsCollectionInterval	`Integer (0..65535)`	Displays the statistics collection interval in minutes. Access: read-only Status: mandatory	`0`
statsBucketInterval	`Integer (0..65535)`	Displays the statistics bucket interval in minutes. Access: read-only Status: mandatory	`0`
userName	`DisplayString {SIZE (1..20)}`	Displays the username of the person logging in or logging out. It is used only in traps, and is currently limited to 12 characters. Access: read-only Status: mandatory	none
shelfIntegratedAlarm	`Integer {clear (1), minor (2), major (3), critical (4)}`	Displays the alarm status of the shelf. An implementation cannot support all the values. The following values are possible: • `clear (1)`—Indicates that the shelf is not in alarm. • `minor (2)`— Indicates that a nonservice affecting condition occurred and that corrective action is being taken to prevent a more serious fault. • `major (3)`—Indicates that a service-affecting condition has occurred and urgent corrective action is required. • `critical (4)`—Indicates that a service effecting condition occurred and immediate corrective action is required. **Note** The definitions for `minor (2)`, `major (3)`, and `critical (4)` are taken from Bellcore GR-1248-CORE. Access: read-only Status: mandatory	`clear`

axisRedundancy Group

This group contains information about redundancy mapping. The Core Card Set in the Cisco MGX 8850 includes the Processor Switch Module (PXM) and the Service Redundancy/Resource Module (System Resource Manager).

Note The slot numbers that are plugged in for the Core Card Set are different for various products.

A redundant pair is a set of cards or modules that operate in a redundant manner. At any time, one or more cards or modules of the pair are active, while the other is standby. If an active member fails, one of the standby members becomes active.

The object identifier for each MIB object is listed in Table 3-144.

Table 3-144 axisRedundancy Group Object Identifiers

Name	Object Identifier
smRedMapTable	::= { axisRedundancy 1 }
smRedMapEntry	::= { smRedMapTable 1}
redPrimarySlotNum	::= { smRedMapEntry 1 }
redRowStatus	::= { smRedMapEntry 2 }
redPrimaryType	::= { smRedMapEntry 3 }
redPrimaryState	::= { smRedMapEntry 4 }
redSecondarySlotNum	::= { smRedMapEntry 5 }
redSecondaryType	::= { smRedMapEntry 6 }
redSecondaryState	::= { smRedMapEntry 7 }
redType	::= { smRedMapEntry 8 }
redCoveringSlot	::= { smRedMapEntry 9 }
redFeature	::= { smRedMapEntry 10 }
redLineModuleType	::= { smRedMapEntry 11 }

The MIB objects are listed in Table 3-145.

Table 3-145 axisRedundancy Group MIB Objects

Name	Syntax	Description	Default Value
smRedMapTable	`Sequence Of SmRedMapEntry`	Provides the redundancy map for a service module. This table is used to configure redundancy for service modules. It contains entries for the Core Card Set in some implementations. Access: not-accessible Status: mandatory	none
s**mRedMapEntry**	`smRedMapEntry`	Provides an entry for Service Module or Core Card Set redundancy. Some implementations can have entries for the core-card set. The entries for core-card set can not be added, modified, or deleted. An entry is created in this table to configure service modules in redundant configuration. Following are examples: • 1:1 (Y cable) redundancy configuration—Exists for each redundant pair with `redPrimarySlotNum` as the index. • 1:N redundancy configuration—Exists with different primary slot numbers (`redPrimarySlot` value) with the same secondary slot numbers (`redSecondarySlot` value). Access: not-accessible Status: mandatory The index contains `redPrimarySlotNum`.	none
redPrimarySlotNum	`Integer (1..32)`	Describes the index to the table. This is the slot number of the primary module in the redundant configuration. For systems where the entries are created by the agent for the Core Card Set, the implied slot numbers are used for this object. The implied slot number values depend on the system. Access: read-write Status: mandatory	none

Table 3-145 axisRedundancy Group MIB Objects (continued)

Name	Syntax	Description	Default Value
redRowStatus	`Integer {add(1), del(2), mod(3)}`	Adds, deletes, or modifies the redundancy. **Note** Modify is not used, but is there for consistency. This object creates and deletes an entry in `smRedMapTable`. This object is set to `add (1)`, to create an entry in the table. The values of `redPrimarySlot` and `redSecondarySlot` objects constitute the redundant pairs. The slot numbers specified for `redPrimarySlot` and `redSecondarySlot` can be in the same half of the shelf for half-height service modules. This object is set to `del (2)` to delete an entry from the table. Access: read-write Status: mandatory	`del`
redPrimaryType	`Integer {other (1), bsc (2), tim (20) reserved for MGX8850 after 1000 pxm1 (1000), pxm1-2t3e3 (1001), pxm1-4oc3 (1002), pxm1-oc12 (1003), rpm (2000)}`	Holds the type of the primary card. Some of the card types are not supported in the table but they are defined for consistency. The bsc, ausm-T3, and tim modules are not supported. **Note** The `redPrimaryType` object is not supported by PXM45 for Release 2.1. Access: read-only Status: mandatory	none
redPrimaryState	`Integer{nocard (1), standby (2), active (3), failed (4), selfTest (5), heldInReset (6), boot (7), mismatch (8), unknown (9), unusedCoreCardMisMatch (10), blocked (11), reserved (12), Hold state applies to PXM during Graceful Upgrade, unusedHold (13)}`	Holds the state of the primary card. Some of the states are not supported in the table but are defined here for consistency with the card states. The values `heldInReset`, `boot`, and `unknown` are not supported. Access: read-only Status: mandatory The following are the supported values: • `nocard (1)`—Module not present in the slot. • `standby (2)`—Module is in standby state. • `active (3)`—Module is in active state. • `failed (4)`—Module is in failed state due to some condition. • `selfTest (5)`—Module is performing self test. • `mismatch(8)`—Wrong module is plugged in.	none

Table 3-145 axisRedundancy Group MIB Objects (continued)

Name	Syntax	Description	Default Value
redSecondarySlotNum	`Integer (0..32)`	Contains the slot number of the redundant card for the primary card. The value of `SecondarySlotNum` is 0 when no secondary card covers the primary card. The value of `SecondarySlotNum` is from the same half of the shelf. If the primary card is in the top half, the secondary card is in the same half and is covered by the top SRM. Access: read-write Status: mandatory	none

redSecondaryType	`Integer {other (1),`
	`bsc (2),`
	`tim (20), frsm-4T1 (30),`
	`reserved for MGX8850 after`
	`1000 pxm1 (1000),`
	`pxm1-2t3e3 (1001),`
	`pxm1-4oc3 (1002),`
	`pxm1-oc12 (1003),`
	`rpm (2000)}`
		Holds the type of the redundant card.
		Note `redSecondaryType` is the same as `redPrimaryType` unless it is changed for a new service module.
		Some of the card types are not supported in the table but they are defined here for consistency with the shelf card types. The bsc, ausm-T3, and tim modules are not supported.
		Note The `redSecondaryType` object is not supported by PXM45 for Release 2.1.
		Access: read-only
		Status: mandatory
			none

redSecondaryState	`Integer {nocard (1),`
	`standby (2),`
	`active (3),`
	`failed (4),`
	`selfTest (5),`
	`heldInReset (6),`
	`boot (7),`
	`mismatch (8),`
	`unknown (9),`
	`unusedCoreCardMisMatch (10),`
	`blocked (11),`
	`reserved (12),`
	`Hold state applies to PXM`
	`during Graceful Upgrade`
	`unusedHold (13)}`
		Holds the state of the secondary card.
		Access: read-only
		Status: mandatory
		The following are the supported values:
		• `nocard (1)`—Module not present in the slot.
		• `standby (2)`—Module is in standby state.
		• `active (3)`—Module is in active state.
		• `failed (4)`—Module is in failed state.
		• `selfTest (5)`—Module is performing self test.
		• `mismatch (8)`—Module is not compatible with the current configuration or wrong type of backcard. A line module is inserted.
			none

Table 3-145 axisRedundancy Group MIB Objects (continued)

Name	Syntax	Description	Default Value
redType	`Integer{yCable (1), oneToN (2) }`	Describes the type of redundancy. The supported values are either `yCable (1)` or `oneToN (2)`. `yCable` is a 1:1 (y cable) redundancy configuration. In this configuration, there is only one module, which acts as the backup for the other. `oneToN (2)` is a 1:N redundancy configuration. In this configuration, there is one module (secondary), which acts as the backup for other primary modules. There are multiple primary modules and one secondary module in this configuration. At any point, the secondary module backsup only one failed primary module. If more than one primary module fails, the failed primary modules are not covered by the secondary module. Access: read-write Status: mandatory	none
redCoveringSlot	`Integer (0..32)`	Indicates the slot number of the primary card. The secondary card is covering this slot number. **Note** `redCoveringSlot` is set to 0 when the primary is not being protected. If this primary card is being protected by its redundant card, this object is the primary slot number. Access: read-only Status: mandatory	none

Table 3-145 axisRedundancy Group MIB Objects (continued)

Name	Syntax	Description	Default Value
redFeature	`Integer`	Indicates the feature of the primary card. Access: read-only Status: mandatory	none
redLineModuleType	`Integer {lm-DB15-4T1 (16), lm-DB15-4E1 (17), lm-BNC-4E1 (18), lm-DB15-4T1-R(19), lm-DB15-4E1-R(20), lm-BNC-4E1-R (21), lm-RJ48-8T1 (22), lm-RJ48-8E1 (23), lm-SMB-8E1 (24), lm-RJ48-T3T1 (25), lm-RJ48-E3E1 (26), lm-RJ48-T3E1 (27), lm-SMB-E3E1 (28), lm-RJ48-E3T1 (29), lm-SMB-T3E1 (30), lm-T3E3-D (32), lm-T3E3-B (33), lm-RJ48-8T1-R(48), lm-RJ48-8E1-R(49), lm-SMB-8E1-R (50), HSSI/X.21 lm-HS1-4X21 (60), lm-HS1-3HSSI(61), HSSI/X.21 lm-HS1-4V35 (62)}`	Indicates the line module type for the primary card. This object is not configured and it has only the type of the current line module that is present in the slot. **Note** The types that end with a -R are a redundant back card type, which do not have a connector on the faceplate. They are used for the redundant card but they are plugged into the primary by mistake. This object shows the applicable type. **Note** The `redLineModuleType` object is not supported by PXM45 for Release 2.1. Access: read-only Status: mandatory	none

Generic MIB for Traps

This section describes the MIB object that make up the GENERICOBJECT-MIB.my file.

Note The Generic MIB for Traps support PXM45 and defined under the StrataCom Enterprise.

The generic objects group is used generically for traps. The objects cannot be read or set.

Note Only the `genericTimeStamp` object is supported.

One object identifier is listed in Table 3-146.

Table 3-146 Generic Object MIB Object Identifier

Name	Object Identifier
genericTimeStamp	::= { genericObjects 3}

One MIB object is listed in Table 3-147.

Table 3-147 Generic Object MIB

Name	Syntax	Description	Default Value
genericTimeStamp	`DisplayString (SIZE (1..30))`	Specifies the generic time stamp used in traps. This object cannot be set or read. The `genericTimeStamp` object is used by all new traps to tell the time that a given trap was originated. Access: read-only Status: mandatory	none

Module MIB for Traps

This section describes the MIB object that make up the BASIS-GENERIC-MIB.my file from the AXIPOP-MIB.

Note The Module MIB for Traps support PXM45 and defined under the StrataCom Enterprise.

One object identifier is listed in Table 3-148 for the Basis Generic MIB.

Table 3-148 Basis Generic MIB Object Identifier

Name	Object Identifier
moduleTrapAlarmSeverity	::= { cardInformation 15 }

One MIB object is listed in Table 3-149.

Table 3-149 Basis Generic MIB Object

Name	Syntax	Description	Default Value
moduleTrapAlarmSeverity	`INTEGER { minor (1), major (2), dontCare (3), critical (4), error (5), warning (6), notice (7), info (8)}`	Specifies this varbind is sent to managers as part of all `Trap` PDUs to determine the module alarm severity. An implementation cannot support all the possible values. The following are the values: • `major (1)`—Specifies major service has been impacted. • `minor (2)`—Specifies minor service has been lost. • `dontCare(3)`—Specifies severity is not applicable. • `critical (4)`—Affects existing data traffic. • `error (5)`—Specifies an error has occurred. • `warning (6)`—Specifies a threshold has been reached. • `notice (7)`—Specifies a normal but significant event has occurred. • `info(8)`—Specifies the applicable information. Access: read-only Status: mandatory	none

Service Class Template

This section describes the MIB objects that make up the CISCO-WAN-SCT-MIB.my file. The Service Class Template (SCT) MIB presents the software configurable parameters of the hardware. The parameters are used to implement various Quality of Service (QoS) and policing features for various service types.

Note The SCT MIB supports both AXSM and AXSM-E.

The SCT MIB groups include:

- Version Number Used for the Service Class Template
- Virtual Circuit Descriptor Group
- Class of Service Buffer Descriptor Group
- Service Class Template Conformance Information

Version Number Used for the Service Class Template

One object identifier is listed in Table 3-150.

Table 3-150 Version Number Object Identifier

Name	Object Identifier
cwSctVersion	::= { cwSctGeneral 1 }

One MIB object is listed in Table 3-151.

Table 3-151 Version Number Used for the Service Class Template MIB Object

Name	Syntax	Description	Default Value
cwSctVersion	`Integer32 (1..65535)`	Specifies the version of the SCT file. Max-Access: read-only Status: current	none

Virtual Circuit Descriptor Group

The object identifiers that are used for the Virtual Circuit (VC) descriptor group are listed in Table 3-152.

Table 3-152 Virtual Circuit Descriptor Group Object Identifiers

Name	Object Identifier
cwSctVcDescTable	::= { cwSctVcDesc 1 }
cwSctVcDescEntry	::= { cwSctVcDescTable 1 }
cwSctId	::= { cwSctVcDescEntry 1 }
cwSctVcDescServiceType	::= { cwSctVcDescEntry 2 }
cwSctVcDescServiceCategory	::= { cwSctVcDescEntry 3 }
cwSctVcDescCosbNumber	::= { cwSctVcDescEntry 4 }
cwSctVcDescCacTreatment	::= { cwSctVcDescEntry 5 }
cwSctVcDescUpcEnable	::= { cwSctVcDescEntry 6 }
cwSctVcDescUpcClpSelection	::= { cwSctVcDescEntry 7 }
cwSctVcDescPolicingActGcra1	::= { cwSctVcDescEntry 8 }
cwSctVcDescPolicingActGcra2	::= { cwSctVcDescEntry 9 }
cwSctVcDescPcr	::= { cwSctVcDescEntry 10 }
cwSctVcDescScr	::= { cwSctVcDescEntry 11 }
cwSctVcDescMcr	::= { cwSctVcDescEntry 12 }
cwSctVcDescIcr	::= { cwSctVcDescEntry 13 }
cwSctVcDescMbs	::= { cwSctVcDescEntry 14 }

Table 3-152 Virtual Circuit Descriptor Group Object Identifiers (continued)

Name	Object Identifier
cwSctVcDescMfs	::= { cwSctVcDescEntry 15 }
cwSctVcDescCdvt	::= { cwSctVcDescEntry 16 }
cwSctVcDescVcPktDiscdMode	::= { cwSctVcDescEntry 17 }
cwSctVcDescMaxThreshold	::= { cwSctVcDescEntry 18 }
cwSctVcDescClp1HighThreshold	::= { cwSctVcDescEntry 19 }
cwSctVcDescClp1LowOrEpd1Thresh	::= { cwSctVcDescEntry 20 }
cwSctVcDescEpd0Threshold	::= { cwSctVcDescEntry 21 }
cwSctVcDescEfciThreshold	::= { cwSctVcDescEntry 22 }
cwSctVcDescCosScalingClass	::= { cwSctVcDescEntry 23 }
cwSctVcDescLogicalPortScaling	::= { cwSctVcDescEntry 24 }
cwSctVcDescCiControl	::= { cwSctVcDescEntry 25 }
cwSctVcDescCrmCells	::= { cwSctVcDescEntry 26 }
cwSctVcDescVsvd	::= { cwSctVcDescEntry 27 }
cwSctVcDescAdtf	::= { cwSctVcDescEntry 28 }
cwSctVcDescRdf	::= { cwSctVcDescEntry 29 }
cwSctVcDescRif	::= { cwSctVcDescEntry 30 }
cwSctVcDescNrm	::= { cwSctVcDescEntry 31 }
cwSctVcDescTrm	::= { cwSctVcDescEntry 32 }
cwSctVcDescCdf	::= { cwSctVcDescEntry 33 }
cwSctVcDescTbe	::= { cwSctVcDescEntry 34 }
cwSctVcDescFrtt	::= { cwSctVcDescEntry 35 }
cwSctVcDescWfqEnable	::= { cwSctVcDescEntry 36 }

The MIB objects are listed in Table 3-153.

Table 3-153 Virtual Circuit Descriptor Group MIB Objects

Name	Syntax	Description	Default Value
cwSctVcDesc Table	`SEQUENCE OF CwSctVcDescEntry`	Provides the SCT VC descriptor table. Max-Access: not-accessible Status: current	none
cwSctVcDesc Entry	`CwSctVcDescEntry`	Provides an entry to the SCT VC Descriptor Table. Max-Access: not-accessible Status: current The following are the indexes: • `cwSctId` • `cwSctVcDescServiceType`	none

Table 3-153 Virtual Circuit Descriptor Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cwSctId	Integer32 (1..65535)	Determines the SCT identifier. Max-Access: not-accessible Status: current	none
cwSctVcDesc ServiceType	Integer32 (0..65535)	Specifies the service type characteristics that are defined by the rest of the parameters. Max-Access: not-accessible Status: current	none
cwSctVcDesc ServiceCategory	Integer32 (0..65535)	Determines the location of the service category to the service type. All service types that belong to the same service category should be mapped to the same Class of Service Buffer (CoSB). Max-Access: read-only Status: current	none
cwSctVcDesc CosbNumber	Integer32 (1..16)	Associates the CoSB number with the service type. Max-Access: read-write Status: current	1
cwSctVcDescCac Treatment	Integer32 (1..256)	Specifies the Connection Admission Control (CAC) algorithm to use with this service type. Max-Access: read-write Status: current The following are the CAC algorithms:	2

Value	Type
1	IcnCac
2	basicCac
3	eCac-Model A
4	eCac-Model B
5	eCac-Model C
6	eCac-Model D
7	eCac-Model E
8	eCac-Model F
9	mbBwCac

Table 3-153 Virtual Circuit Descriptor Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cwSctVcDescUpc Enable	INTEGER { enableAll(1), enableGcra1(2), enableGcra2(3), enableGcra1WithPktPolicing(4), enableGcra2WithPktPolicing(5), disableAll(6) }	Enables or disables the selective Usage Parameter Control (UPC) policing on the virtual circuit. Max-Access: read-write Status: current	none
cwSctVcDescUpc ClpSelection	Integer32 (1..4)	Selects the processing of policing buckets based on the Cell Loss Priority (CLP) bit. Max-Access: read-write Status: current	none
cwSctVcDesc PolicingActGcra1	INTEGER { discard(1), setClpBit(2), setClpDiscTagged(3) }	Determines the type of policing action the policer must take for the cells in bucket 1 for this service type. Max-Access: read-write Status: current	none
cwSctVcDesc PolicingActGcra2	INTEGER { discard(1), setClpBit(2), setClpDiscTagged(3) }	Determines the type of policing action the policer must take for the cells in bucket 2 for this service type. Max-Access: read-write Status: current	none
cwSctVcDescPcr	Integer32 (0..1000000)	Specifies the PCR for any connection this service type is allowed to send. The cwSctVcDescPcr object represents a percentage of the logical interface maximum rate. A value of 1000000 corresponds to 100%. Max-Access: read-write Status: current	none
cwSctVcDescScr	Integer32 (0..1000000)	Specifies that the Sustained Cell Rate (SCR) represents a percentage of the PCR. A value of 1000000 corresponds to 100%. Max-Access: read-write Status: current	none
cwSctVcDescMcr	Integer32 (0..1000000)	Specifies that the Minimum Cell Rate (MCR) represents a percentage of the Peak Cell Rate. A value of 1000000 corresponds to 100%. Max-Access: read-write Status: current	none

Table 3-153 Virtual Circuit Descriptor Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cwSctVcDescIcr	`Integer32 (0..1000000)`	Specifies that the Initial Cell Rate (ICR) is used for ABR connections that are idle. The `cwSctVcDescIcr` object represents a percentage of the peak cell rate. A value of 1000000 corresponds to 100%. Max-Access: read-write Status: current	none
cwSctVcDescMbs	`Integer32 (1..5000000)`	Specifies that the Maximum Burst Size (MBS) represents a policing parameter. Max-Access: read-write Status: current	none
cwSctVcDescMfs	`Integer32 (1..1236)`	Specifies that the Maximum Frame Size (MFS) is used for frame based policing. MFS supports 64Kb that equals to approximately 1236 cells. Max-Access: read-write Status: current	none
cwSctVcDesc Cdvt	`Integer32 (0..5000000)`	Specifies that the Cell Delay Variation Tolerance (CDVT) is used for policing purposes. The `cwSctVcDescCdvt` object is represented in microseconds. Max-Access: read-write Status: current	none
cwSctVcDesc VcPktDiscdMode	`INTEGER { enable(1), disable(2) }`	Enables or disables the packet discard mode for the VC. If the packet mode is enabled, the Early Packet Discard (EPD) threshold is activated. If the packet mode disabled, CLP thresholds are active. Max-Access: read-write Status: current	none
cwSctVcDescMax Threshold	`Integer32 (0..5000000)`	Specifies that the VC maximum threshold indicates the maximum number of cells that are buffered in the cell memory for that VC. It is represented in units of time in ms. The value is multiplied by the line rate that gives the number of cells for the threshold. Max-Access: read-write Status: current	none
cwSctVcDescClp1 HighThreshold	`Integer32 (0..1000000)`	Specifies that the CLP1 high threshold represents a percentage of the VC maximum threshold. A value of 1000000 corresponds to 100%. Max-Access: read-write Status: current	none

Table 3-153 Virtual Circuit Descriptor Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cwSctVcDesc Clp1LowOrEpd1 Thresh	`Integer32 (0..1000000)`	Specifies that the contents of this field is treated as an early Packet Discard (EPD1) threshold if the packet discard mode is on. If not, the contents are treated as a CLP1 low threshold. CLP1 low threshold represents a percentage of the CoSB maximum threshold. A value of 1000000 corresponds to 100%. The EPD1 threshold represents a percentage of the CoSB maximum threshold. A value of 1000000 corresponds to 100%. Max-Access: read-write Status: current	none
cwSctVcDescEpd0 Threshold	`Integer32 (0..1000000)`	Specifies that the EPD threshold represents a percentage of the CoSB maximum threshold. If the packet mode is on, this corresponds to EPD (0+1) threshold. A value of 1000000 corresponds to 100%. Max-Access: read-write Status: current	none
cwSctVcDescEfci Threshold	`Integer32 (0..1000000)`	Specifies that the Explicit Forward Congestion Indication (EFCI) threshold is a percentage of the VC maximum cell threshold. A value of 1000000 corresponds to 100%. Max-Access: read-write Status: current	none
cwSctVcDescCos ScalingClass	`Integer32 (1..4)`	Specifies that the scaling class reduces the maximum queue depth when the service group or global cell memory congestion is experienced. Max-Access: read-write Status: current	none
cwSctVcDesc LogicalPortScaling	`Integer32 (1..4)`	Specifies that the logical port scaling class reduces the maximum queue depth in the arrival engine for the logical port. Max-Access: read-write Status: current	none
cwSctVcDescCi Control	`INTEGER { enable(1), disable(2) }`	Enables or disables the congestion indication setting in the egress Resource Management (RM) cells if the EFCI threshold is exceeded. Max-Access: read-write Status: current	none

Table 3-153 Virtual Circuit Descriptor Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cwSctVcDescCrm Cells	`Integer32 (1..4095)`	Cuts off the RM cells. The ABR maximum number of forwarding ingress RM cells are sent in absence of the backward RM cells. Max-Access: read-write Status: current	none
cwSctVcDescVsvd	`INTEGER { enableWithFCES(1), enableWithoutFCES(2), disable(3) }`	Specifies the ABR VS/VD indicator.The `Get` operation for this object is valid only for ABRSTD connections. **Note** This object is also used as a fast end point trigger to allow for proper congestion setup. Max-Access: read-write Status: current	none
cwSctVcDescAdtf	`Integer32 (1..1023)`	Decreases the time factor for ACR that is the idle time before ACR to ICR. Units: 10 ms Max-Access: read-write Status: current	none
cwSctVcDescRdf	`INTEGER { rdfOneOver32768(1), rdfOneOver16384(2), rdfOneOver8192(3), rdfOneOver4096(4), rdfOneOver2048(5), rdfOneOver1024(6), rdfOneOver512(7), rdfOneOver256(8), rdfOneOver128(9), rdfOneOver64(10), rdfOneOver32(11), rdfOneOver16(12), rdfOneOver8(13), rdfOneOver4(14), rdfOneOver2(15), rdfOne(16) }`	Specifies the ABR rate decrease factor. Max-Access: read-write Status: current	none

Table 3-153 Virtual Circuit Descriptor Group MIB Objects (continued)

Name	Syntax	Description	Default Value

cwSctVcDescRif	INTEGER
	{
	rifOneOver32768(1),
	rifOneOver16384(2),
	rifOneOver8192(3),
	rifOneOver4096(4),
	rifOneOver2048(5),
	rifOneOver1024(6),
	rifOneOver512(7),
	rifOneOver256(8),
	rifOneOver128(9),
	rifOneOver64(10),
	rifOneOver32(11),
	rifOneOver16(12),
	rifOneOver8(13),
	rifOneOver4(14),
	rifOneOver2(15),
	rifOne(16)
	}
		Specifies the ABR rate increase factor.
		Max-Access: read-write
		Status: current
			none

cwSctVcDescNrm	INTEGER
	{
	nrm2(1),
	nrm4(2),
	nrm8(3),
	nrm16(4),
	nrm32(5),
	nrm64(6),
	nrm128(7),
	nrm256(8)
	}
		Determines the number of data cells between frame cells.
		Max-Access: read-write
		Status: current
			none

cwSctVcDescTrm	INTEGER
	{
	trm0point78125(1),
	trm1point5625(2),
	trm3point125(3),
	trm6point25(4),
	trm12point5(5),
	trm25(6),
	trm50(7),
	trm100(8)
	}
		Determines the time between forward RM cells. The value Trm = 100/(2 ^ RTrm). For example, RTrm is between the range of 0 to 7. The value provided is for RTrm.
		Max-Access: read-write
		Status: current
			none

Table 3-153 Virtual Circuit Descriptor Group MIB Objects (continued)

Name	Syntax	Description	Default Value

cwSctVcDescCdf	INTEGER
	{
	cdf0(1),
	cdfOneOver64(2),
	cdfOneOver32(3),
	cdfOneOver16(4),
	cdfOneOver8(5),
	cdfOneOver4(6),
	cdfOneOver2(7),
	cdfOne(8)
	}
		Cuts off the decrease factor.
		Max-Access: read-write
		Status: current
			none

cwSctVcDescTbe	Integer32 (0..16777215)	Specifies the transient buffer exposure. Max-Access: read-write Status: current	none
cwSctVcDescFrtt	Integer32 (0..16700000)	Specifies the fixed round trip time. Units: time in ms Max-Access: read-write Status: current	none

cwSctVcDescWfq Enable	INTEGER
	{
	enable(1),
	disable(2)
	}
		Enables or disables the weighted fair queueing per VC.
		Max-Access: read-write
		Status: current
			none

Class of Service Buffer Descriptor Group

The object identifiers that are used for the Class of Service Buffer (CoSB) descriptor group are listed in Table 3-154.

Table 3-154 Class of Service Buffer Descriptor Group Object Identifiers

Name	Object Identifier
cwSctCosbTable	::= { cwSctCosbDesc 1 }
cwSctCosbEntry	::= { cwSctCosbTable 1 }
cwSctCosbSctId	::= { cwSctCosbEntry 1 }
cwSctCosbNumber	::= { cwSctCosbEntry 2 }
cwSctCosbMinRate	::= { cwSctCosbEntry 3 }
cwSctCosbMaxReservableRate	::= { cwSctCosbEntry 4 }
cwSctCosbMinPriority	::= { cwSctCosbEntry 5 }
cwSctCosbExcessPriority	::= { cwSctCosbEntry 6 }
cwSctCosbMaxThreshold	::= { cwSctCosbEntry 7 }

Table 3-154 Class of Service Buffer Descriptor Group Object Identifiers (continued)

Name	Object Identifier
cwSctCosbClp1HighThreshold	::= { cwSctCosbEntry 8 }
cwSctCosbClp1LowOrEpd1Threshold	::= { cwSctCosbEntry 9 }
cwSctCosbEpd0Threshold	::= { cwSctCosbEntry 10 }
cwSctCosbEfciThreshold	::= { cwSctCosbEntry 11 }
cwSctCosbErs	::= { cwSctCosbEntry 12 }
cwSctCosbRedSelection	::= { cwSctCosbEntry 13 }
cwSctCosbRedThreshold	::= { cwSctCosbEntry 14 }
cwSctCosbRedProbabilityFactor	::= { cwSctCosbEntry 15 }
cwSctCosbWfq	::= { cwSctCosbEntry 16 }
cwSctCosbBestEffortIndicator	::= { cwSctCosbEntry 17 }
cwSctCosbDiscardAlarmEnable	::= { cwSctCosbEntry 18 }
cwSctCosbDiscardAlarmThresh	::= { cwSctCosbEntry 19 }
cwSctCosbCellLossRatio	::= { cwSctCosbEntry 20 }

The MIB objects are listed in Table 3-155.

Table 3-155 Class of Service Buffer Descriptor Group MIB Objects

Name	Syntax	Description	Default Value
cwSctCosbTable	`SEQUENCE OF CwSctCosbEntry`	Provides the Service Class Template CoSB descriptor table. Max-Access: not-accessible Status: current	none
cwSctCosbEntry	`CwSctCosbEntry`	Provides an entry in the Service Class Template CoSB descriptor table. Max-Access: not-accessible Status: current The following are the indexes: • `cwSctCosbSctId` • `cwSctCosbNumber`	none
cwSctCosbSctId	`Integer32 (1..65535)`	Provides the service class template identifier. Max-Access: not-accessible Status: current	none
cwSctCosbNumber	`Integer32 (1..16)`	Indicates that he CoSB properties are defined by the rest of the fields. Max-Access: not-accessible Status: current	none

Table 3-155 Class of Service Buffer Descriptor Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cwSctCosbMin Rate	`Integer32 (0..1000000)`	Indicates the minimum or the guaranteed bandwidth allocated for the CoSB. The bandwidth represents a percentage of the logical interface minimum rate. A value of 1000000 is equal to 100%. Max-Access: read-write Status: current	none
cwSctCosbMax ReservableRate	`Integer32 (0..1000000)`	Indicates the maximum guaranteed bandwidth this CoSB can have at any time. A value of 1000000 is equal to 100%. Max-Access: read-write Status: current	none
cwSctCosbMin Priority	`Integer32 (0..15)`	Indicates the priority where the CoSB is serviced to guarantee the bandwidth requirement. If the priority is higher, the CDV performance is better. The value of zero is the highest priority. Max-Access: read-write Status: current	none
cwSctCosbExcess Priority	`Integer32 (0..15)`	Indicates the priority where the CoSB is given access to the excess bandwidth that is available in the service group. The value of zero is the highest priority. Max-Access: read-write Status: current	none
cwSctCosbMax Threshold	`Integer32 (0..5000000)`	Indicates the maximum delay for any CLP (0+1) cell entering this CoSB. Any cell that encounters a queue latency greater than this is dropped. The cells are represented in units of time. Max-Access: read-write Status: current	none
cwSctCosbClp1 HighThreshold	`Integer32 (0..1000000)`	Specifies the CLP low threshold for CLP hysteresis. Any cell that encounters a queue latency greater than this is dropped. The cells are represented as a percentage of the maximum threshold. The value of 1000000 corresponds to 100%. Max-Access: read-write Status: current	none

Table 3-155 Class of Service Buffer Descriptor Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cwSctCosbClp1LowOrEpd1Threshold	`Integer32 (0..1000000)`	Indicates the maximum delay for any CLP(1) cell entering the CoSB. Any cell that encounters a queue latency greater than this is dropped. The cells are represented as a percentage of the maximum threshold. If the packet mode is enabled for the VC, this is created as EPD1 to represent a percentage of the maximum threshold. The value of 1000000 corresponds to 100%. Max-Access: read-write Status: current	none
cwSctCosbEpd0Threshold	`Integer32 (0..1000000)`	Indicates the maximum delay for any CLP (0+1) cell entering the CoSB. Any cell that encounters a queue latency greater than this is dropped. The cells are used in packet mode and is represented as a percentage of the maximum threshold. The value of 1000000 corresponds to 100%. Max-Access: read-write Status: current	none
cwSctCosbEfciThreshold	`Integer32 (0..1000000)`	Indicates the threshold for the EFCI to represent a percentage of the maximum threshold.The value of 1000000 corresponds to 100%. Max-Access: read-write Status: current	none
cwSctCosbErs	`INTEGER { enable(1), disable(2) }`	Indicates whether the explicit rate stamping is enabled or not. Max-Access: read-write Status: current	none
cwSctCosbRedSelection	`INTEGER { enable(1), disable(2) }`	Enables or disables the Random Early Discard (RED). Max-Access: read-write Status: current	none
cwSctCosbRedThreshold	`Integer32 (0..1000000)`	Indicates the threshold for EPD to represent a percentage of the cell maximum threshold. The value of 1000000 corresponds to 100%. Max-Access: read-write Status: current	none
cwSctCosbRedProbabilityFactor	`Integer32 (0..15)`	Indicates the RED probability factor to represent a percentage of the cell maximum threshold. Max-Access: read-write Status: current	none

Table 3-155 Class of Service Buffer Descriptor Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cwSctCosbWfq	INTEGER { enable(1), disable(2) }	Indicates whether the Weighted Fair Queueing (WFQ) is enabled or not. Max-Access: read-write Status: current	none
cwSctCosbBest EffortIndicator	INTEGER { enable(1), disable(2) }	Indicates the treatment for the maximum cell rate on the CoSB. If set, the maximum rate is limited by the service group maximum rate. Max-Access: read-write Status: current	none
cwSctCosbDiscard AlarmEnable	INTEGER { enable(1), disable(2) }	Enables or disables the discard alarm per VC. Max-Access: read-write Status: current	none
cwSctCosbDiscard AlarmThresh	Integer32 (0..10000000)	Specifies if the number of cells discarded in a CoSB exceeds this threshold, a software alarm is generated. If DiscardAlarm is enabled, the software alarm is generated. Max-Access: read-write Status: current	none
cwSctCosbCellLoss Ratio	Integer32 (1..15)	Indicates the Cell Loss Ratio (CLR) that is supported in the CoSB. The actual CLR value is derived as the negative logarithm of this value. Max-Access: read-write Status: current	none

Service Class Template Conformance Information

The object identifiers are listed in Table 3-156 for conformance.

Table 3-156 Service Class Template Conformance Object Identifiers

Name	Object Identifier
ciscoWanSctMIBConformance	::= { ciscoWanSctMIB 3 }
cwSctMIBCompliances	::= { ciscoWanSctMIBConformance 1 }
cwSctMIBGroups	::= { ciscoWanSctMIBConformance 2 }

The MIB objects are listed in Table 3-157.

Table 3-157 Service Class Template MIB Objects Used for Conformance

Name	Mandatory Groups	Description	Default Value
cwSctMIB Compliance	`cwSctGeneralGroup`, `cwSctVcDescMIBGroup`, `cwSctCosbDescMIBGroup`	Provides the compliance statement for the `ciscoSonet` management group. Status: current	none
cwSctVcExtended AbrGroup	none	Specifies that this group is supported when extended ABR features are applicable in the switch.	none
cwSctWfqGroup	none	Specifies that this group is supported only if the Weighted Fair Queueing (WFQ) is evident in the switch for VC and CoSB.	none
cwSctRedGroup	none	Specifies that this group is supported only if the Random Early Discard (RED) is applicable in the switch.	none
cwSctCosbMin Rate	none	Indicates that the percentage of the minimum guaranteed bandwidth of the service group that is guaranteed to the CoSB. Min-Access: not-accessible	none
cwSctCosbMax ReservableRate	none	Indicates that the percentage of the maximum bandwidth of the service group that is allowed for the CoSB. Min-Access: not-accessible	none

Service Class Template Units of Conformance

The object identifiers are listed in Table 3-158 for the units of conformance.

Table 3-158 Service Class Template Units of Conformance

Name	Object Identifier
cwSctGeneralGroup	::= { cwSctMIBGroups 1}
cwSctVcDescMIBGroup	::= { cwSctMIBGroups 2}
cwSctCosbDescMIBGroup	::= { cwSctMIBGroups 3}
cwSctVcExtendedAbrGroup	::= { cwSctMIBGroups 4}
cwSctWfqGroup	::= { cwSctMIBGroups 5}
cwSctRedGroup	::= { cwSctMIBGroups 6}

The MIB objects are listed in Table 3-159 for the units of conformance.

Table 3-159 Service Class Template MIB Objects Used for Conformance

Name	Objects	Description	Default Value
cwSctGeneral Group	cwSctVersion	Determines the collection of objects that represents the general SCT information. Status: current	none
cwSctVcDesc MIBGroup	cwSctVcDescServiceCategory, cwSctVcDescCosbNumber, cwSctVcDescCacTreatment, cwSctVcDescUpcEnable, cwSctVcDescUpcClpSelection, cwSctVcDescPolicingActGcra1, cwSctVcDescPolicingActGcra2, cwSctVcDescPcr, cwSctVcDescScr, cwSctVcDescMcr, cwSctVcDescIcr, cwSctVcDescMbs, cwSctVcDescMfs, cwSctVcDescCdvt, cwSctVcDescVcPktDiscdMode, cwSctVcDescMaxThreshold, cwSctVcDescClp1HighThreshold, cwSctVcDescClp1LowOrEpd1Thresh, cwSctVcDescEpd0Threshold, cwSctVcDescEfciThreshold, cwSctVcDescCosScalingClass, cwSctVcDescLogicalPortScaling, cwSctVcDescCiControl, cwSctVcDescCrmCells	Determines the objects that are used for the VC descriptions. Status: current	none
cwSctCosbDesc MIBGroup	cwSctCosbMinRate, cwSctCosbMaxReservableRate, cwSctCosbMinPriority, cwSctCosbExcessPriority, cwSctCosbMaxThreshold, cwSctCosbClp1HighThreshold, cwSctCosbClp1LowOrEpd1Threshold, cwSctCosbEpd0Threshold, cwSctCosbEfciThreshold, cwSctCosbErs, cwSctCosbBestEffortIndicator, cwSctCosbDiscardAlarmEnable, cwSctCosbDiscardAlarmThresh	Determines the objects that are used for the CoSB descriptions. Status: current	none

Table 3-159 Service Class Template MIB Objects Used for Conformance (continued)

Name	Objects	Description	Default Value
cwSctVcExtended AbrGroup	cwSctVcDescVsvd, cwSctVcDescAdtf, cwSctVcDescRdf, cwSctVcDescRif, cwSctVcDescNrm, cwSctVcDescTrm, cwSctVcDescCdf, cwSctVcDescTbe, cwSctVcDescFrtt	Determines the objects that are used to support the extended ABR features. Status: current	none
cwSctWfqGroup	cwSctVcDescWfqEnable, cwSctCosbWfq	Determines the supported objects if the WFQ is applicable in the switch. Status: current	none
cwSctRedGroup	cwSctCosbRedSelection, cwSctCosbRedProbabilityFactor, cwSctCosbRedThreshold	Determines the supported objects if the RED features are applicable in the switch. Status: current	none

Cisco WAN Class of Service Buffer

This section describes the MIB objects that make up the CISCO-WAN-ATM-COSB-MIB.my file, which manages the parameters related to the Class of Service (CoS) queue and buffer. Every virtual interface contains sixteen CoS queues, which are also known as CoS buffers. Each queue maps to different ATM types of ATM traffic.

Note The Cisco WAN ATM CoSB MIB supports only AXSM-E and is defined under the StrataCom Enterprise.

The Cisco WAN ATM CoSB MIB groups include:

- Interval Statistics Table
- Cisco WAN Class of Service Buffer Conformance Information

Interval Statistics Table

The object identifiers that are used for the Interval Statistics Table are listed in Table 3-160.

Table 3-160 Interval Statistics Table Object Identifiers

Name	Object Identifier
cwacIntervalTable	::= { cwacStatistics 2 }
cwacIntervalEntry	::= { cwacIntervalTable 1 }
cwacIntervalNumber	::= { cwacIntervalEntry 1 }
cwacIntCellArrivals	::= { cwacIntervalEntry 2 }

Table 3-160 Interval Statistics Table Object Identifiers (continued)

Name	Object Identifier
cwacIntCellDiscards	::= { cwacIntervalEntry 3 }
cwacIntCellDeparts	::= { cwacIntervalEntry 4 }

The MIB objects are listed in Table 3-161.

Table 3-161 Interval Statistics Table MIB Objects

Name	Syntax	Description	Default Value
cwacInterval Table	`SEQUENCE OF CwacIntervalEntry`	Reflects the interval statistics associated with each of the CoS queues. When a new port is added, entries in the table are automatically created. Max-Access: not-accessible Status: current	none
cwacInterval Entry	`CwacIntervalEntry`	Determines an entry for each virtual interface CoS queue interval statistics. In addition to the current 15-minute interval bucket, the previous 24 hours worth of 15-minute interval buckets are collected for each virtual interface. The table is indexed by `ifIndex` that belongs to the `ifTable` entry. The `ifType` value is `atmVirtual(149)`. Max-Access: not-accessible Status: current The following are the indexes: • `ifIndex` • `cwacCosbIndex` • `cwacIntervalNumber`	none
cwacInterval Number	`Integer32(0..96)`	Determines a number used to uniquely identify each virtual interface CoS queue interval statistics. The value of 0 identifies the current 15-minute interval. The range of 1 to 96 identifies the previous 24 hours of the 15-minute interval buckets. Max-Access: not-accessible Status: current	none
cwacIntCell Arrivals	`Counter32`	Determines the number of cells that arrived at the queue during a particular 15-minute interval. Max-Access: read-only Status: current	none

Table 3-161 Interval Statistics Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwacIntCell Discards	Counter32	Determines the number of cells discarded due to congestion during a particular 15-minute interval. Max-Access: read-only Status: current	none
cwacIntCell Departs	Counter32	Determines the number of cells that left the QBIN during a particular 15-minute interval. Max-Access: read-only Status: current	none

Cisco WAN Class of Service Buffer Conformance Information

The object identifiers are listed in Table 3-162 for conformance.

Table 3-162 Cisco WAN Class of Service Buffer Conformance Object Identifiers

Name	Object Identifier
ciscoWanAtmCosbMIBConformance	::= { ciscoWanAtmCosbMIB 2 }
ciscoWanAtmCosbMIBCompliances	::= { ciscoWanAtmCosbMIBConformance 1 }
ciscoWanAtmCosbMIBGroups	::= { ciscoWanAtmCosbMIBConformance 2 }

The MIB objects are listed in Table 3-163.

Table 3-163 Cisco WAN Class of Service Buffer MIB Objects Used for Conformance

Name	Mandatory Groups	Description	Default Value
ciscoWanAtmCosbMIBCompliance	ciscoWanAtmCosbAlarmMIBGroup	Specifies the compliance statement for the CoS queue configuration group. Status: current	none
cwacCosbCurrentCellsDiscThres	none	Specifies that write access is optional if thresholds are set through some other mechanism. Min-Access: read-only	none

Cisco WAN Class of Service Buffer Units of Conformance

One object identifier is listed in Table 3-164 for the units of conformance.

Table 3-164 Cisco WAN Class of Service Buffer Units of Conformance

Name	Object Identifier
ciscoWanAtmCosbAlarmMIBGroup	::= { ciscoWanAtmCosbMIBGroups 1 }

One MIB object is listed in Table 3-165 for the units of conformance.

Table 3-165 Cisco WAN Class of Service Buffer MIB Objects Used for Conformance

Name	Objects	Description	Default Value
ciscoWanAtmCosbAlarmMIBGroup	`cwacCosbCurrentCellsDiscThres`, `cwacStatsAlarmStatus`, `cwacValidIntervals`, `cwacIntCellArrivals`, `cwacIntCellDiscards`, `cwacIntCellDeparts`	Determines the statistics for each virtual interface in the CoS queue alarm group. Status: current	none

CHAPTER 4

PXM MIB Objects

This chapter describes the Management Information Base (MIB) objects used for PXM45.

Contents of this chapter include:

- Cisco Entity FRU Control MIB Objects
- Entity MIB Objects
- Sensor MIB Objects
- SNMPv2 MIB Objects
- Cisco VSI Controller MIB Objects

Cisco Entity FRU Control MIB Objects

This section describes the individual MIB objects that make up the PXM45-specific MIB files. The CISCO Entity Field Replaceable Unit (FRU) Control MIB resides in the ENTITY-MIB.my file. The CISCO Entity FRU CONTROL MIB is used to monitor and configure the operational FRU of the system listed in the ENTITY MIB (RFC 2037) `entPhysicalTable`.

The FRUs include assemblies such as power supplies, fans, processor modules, interface modules, and so forth.

The Cisco Entity FRU Control MIB objects include:

- CISCO ENTITY FRU CONTROL MIB Textual Conventions
- cefc Module Table
- CISCO ENTITY FRU CONTROL MIB Notification
- CISCO ENTITY FRU CONTROL MIB Conformance Information

CISCO ENTITY FRU CONTROL MIB Textual Conventions

The names of the textual conventions are specified in the object syntax.

Table 4-1 lists the textual conventions MIB objects.

Note Textual conventions do not have object identifiers.

Table 4-1 Cisco Entity FRU Control Textual Conventions MIB Objects

Name	Syntax	Description	Default Value
PowerRedundancy Type	`Integer {notsupported (1), redundant (2), combined (3)}`	Contains the valid values of the power supply redundancy modes. The following are the values: • `notsupported(1)`—Indicates that only the requested administrative state (redundant [2] or combined [3]) is not supported by the system. This is the operational state. • `redundant(2)`—Signifies a single power supply output that powers the entire system, although there is more than one supply in the system. For example, a system with one 1000-watt supply and one 1300-watt supply can be rated to power 1000-watts of FRUs. • `combined(3)`—Specifies the combined output of the power supplies that are available to operate the system. For example, two 1000-watt supplies a combined-mode supply of 2000 watts to the system. Status: current	none
PowerAdminType	`Integer {on(1), off(2)}`	Sets the administratively desired FRU power state types. Valid values are the following: • `on (1)` • Turn FRU `on` and `off (2)` • Turn FRU `off` Status: current	none
PowerOperType	`Integer {offEnvOther(1), on(2), offAdmin(3), offDenied(4), offEnvPower(5), offEnvTemp(6), offEnvFan(7)}`	Sets the operational FRU status types. The following are the values: • `offEnvOther(1)`—Specifies that FRU is powered off because of a problem not listed below. • `on(2)`—Specifies that FRU is powered on. • `offAdmin(3)`—Turns administratively off. • `offDenied(4)`—Specifies that FRU is powered off because the available system power is insufficient. • `offEnvPower(5)`—Turns the FRU is powered off because of a power problem. For example, the FRU power translation (DC-DC converter) or distribution failed. • `offEnvTemp(6)`—Turns the FRU power off because of a temperature problem. • `offEnvFan(7)`—Turns the FRU power off because of fan problems. Status: current	none

Table 4-1 Cisco Entity FRU Control Textual Conventions MIB Objects (continued)

Name	Syntax	Description	Default Value
FRUCurrentType	`Integer ( -1000000000..1000000000 )`	Shows a current measurement on the system power supply primary output in `cefcPowerUnits`. Range is from negative 1 million to positive one million amperes. A negative value expresses the current used by the FRU. A positive value expresses the current supplied by the FRU. Status: current	none
ModuleAdminType	`Integer {enabled (1), disabled (2), reset (3), outOfServiceAdmin (4)}`	Sets the administratively desired module states. The following are the values: • `enabled (1)`—Determines that the module is operational. • `disabled (2)`—Determines that the module is not operational. • `reset (3)`—Resets the module. • `outOfServiceAdmin (4)`—Turns the module power on, but it is out of service. This value is set by CLI. Status: current	none
ModuleOperType	`Integer {unknown(1), ok(2), disabled(3), okButDiagFailed(4), boot(5), selfTest(6), failed(7), missing(8), mismatchWithParent(9), mismatchConfig(10), diagFailed(11), dormant(12), outOfServiceAdmin(13), outOfServiceEnvTemp(14)}`	Displays the operational module states. The following are the values: • `unknown(1)`—Specifies that the module is not in one of the other recognized operational states. • `ok(2)`—States the module is operational. • `disabled(3)`—Disables the module administratively. • `okButDiagFailed(4)`—Specifies that the module is operational but there is some diagnostic information available. This is a transitional state. • `boot(5)`—Brings up the image of the module. After boot, it starts its operational software and transitions to the appropriate state. This is a transitional state. • `selfTest(6)`—Performs a self-test of the module. This is a transitional state. • `failed(7)`—Specifies that the module has failed due to some condition not stated above. This is a failure state. • `missing(8)`—Specifies that the module is provisioned, but it is missing. This is a failure state. • `mismatchWithParent(9)`—Indicates that the module is not compatible with parent entity. Module is not provisioned; the wrong type of module is plugged in. This state is cleared by plugging in the appropriate module. This is a failure state.	none

Table 4-1 Cisco Entity FRU Control Textual Conventions MIB Objects (continued)

Name	Syntax	Description	Default Value
ModuleOperType (continued)		• `mismatchConfig(10)`—Indicates that the module is not compatible with the current configuration. The module was provisioned correctly earlier; however, the module has been replaced by an incompatible module. This state is resolved by clearing the configuration or by replacing the appropriate module. This is a failure state. • `diagFailed(11)`—Specifies that the module diagnostic test failed due to a hardware failure. This is a failure state. • `dormant(12)`—Specifies that the module is waiting for an external or internal event to become operational. • `outOfServiceAdmin(13)`—Turns on the module administratively; however, it is out of service. • `outOfServiceEnvTemp(14)`—Turns on the module administratively; however, it is out of service due to an environmental temperature problem. An out-of-service module consumes less power and consequently can cool down. Status: current	
ModuleResetReason Type	`Integer {unknown(1), powerUp(2), parityError(3), clearConfigReset(4), manualReset(5)}`	Describes the reason for the last module reset operation. The following are the values: • `unknown(1)`—Source of the reset is not identified. • `powerUp(2)`—Indicates a system powerup operation. • `parityError(3)`—Indicates a parity error that occurred during the system bring up operation. • `clearConfigReset(4)`—Resets due to clear configuration operation. • `manualReset(5)`—Resets due to an administrative request. Status: current	none

cefc Module Table

The object identifier for each MIB object is listed in Table 4-2.

Table 4-2 cefc Module Table Object Identifiers

Name	Object Identifier
cefcModuleTable	::= { cefcModule 1 }
cefcModuleEntry	::= { cefcModuleTable 1 }
cefcModuleAdminStatus	::= { cefcModuleEntry 1 }
cefcModuleOperStatus	::= { cefcModuleEntry 2 }
cefcModuleResetReason	::= { cefcModuleEntry 3 }
cefcModuleStatusLastChangeTime	::= { cefcModuleEntry 4 }

The MIB objects are listed in Table 4-3.

Table 4-3 cefc Module Table MIB Objects

Name	Syntax	Description	Default Value
cefcModuleTable	`Sequence of cefcModuleEntry`	Lists the operation and administrative status information for ENTITY MIB `entPhysicalTable` entries for the FRUs of type `PhysicalClass module(9)`. Max-Access: not-accessible Status: current	none
cefcModuleEntry	`cefcModuleEntry`	Lists the operation and administrative status information for ENTITY MIB `entPhysicalTable` entries for FRUs of type `PhysicalClass` module (9). Entries are created by the agent at the system power-up or module insertion. Entries are deleted by the agent upon module removal. Max-Access: not-accessible Status: current The index contains `entPhysicalIndex`.	none
cefcModuleAdminStatus	`ModuleAdminType`	Provides administrative control of the module. Max-Access: read-write Status: current	none
cefcModuleOperStatus	`ModuleOperType`	Shows the operational state of the module. Max-Access: read-only Status: current	none

Table 4-3 cefc Module Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cefcModuleResetReason	ModuleResetReasonType	Identifies the reason for the last reset performed on the module. Max-Access: read-only Status: current	none
cefcModuleStatusLast ChangeTime	TimeStamp	Displays the value of sysUpTime when the cefcModuleOperStatus is changed. Max-Access: read-only Status: current	none

CISCO ENTITY FRU CONTROL MIB Notification

According to RFC 2578, a notification is used to describe unsolicited transmissions of management information. NOTIFICATION-TYPE conveys the syntax and semantics of a notification.

The object identifier for the MIB object is listed in Table 4-4.

Table 4-4 CISCO ENTITY FRU CONTROL MIB Notification Object Identifier

Name	Object Identifier
cefcModuleStatusChange	::= { cefcMIBNotifications 1 }

One MIB object is listed in Table 4-5.

Table 4-5 CISCO ENTITY FRU CONTROL MIB Notification MIB Object

Name	Objects	Description	Default Value
cefcModuleStatusChange	cefcModuleOperStatus, cefcModuleStatusLast ChangeTime	Generates the value of cefcModuleOperStatus changes. NMS uses this object to update the status of the module it is managing. Status: current	none

CISCO ENTITY FRU CONTROL MIB Conformance Information

The information on conformance is specific to SNMPv2 as listed in Table 4-6 and Table 4-7.

Table 4-6 CISCO ENTITY FRU CONTROL MIB Units of Conformance

Name	Object Identifier
cefcMIBModuleGroup	::= { cefcMIBGroups 3 }

Table 4-7 CISCO ENTITY FRU CONTROL MIB Units of Conformance Object

Name	Objects	Description	Default Value
cefcMIBModuleGroup	`cefcModuleAdminStatus, cefcModuleOperStatus, cefcModuleResetReason, cefcModuleStatusLast ChangeTime`	Provides a collection of objects that are used to obtain the operational state and redundancy state of the modules. Status: current	none

Entity MIB Objects

This section describes the individual MIB objects that make up the PXM45-specific MIB files. The ENTITY MIB resides in the ENTITY-MIB.my file. This MIB module represents multiple logical entities supported by a single SNMP agent.

The entity MIB objects include:

- Physical Table
- Physical Entity Table
- Physical Mapping Table
- Last Change TimeStamp
- Entity Conformance and Compliance Information

Physical Table

The physical table defines the textual conventions.

Note The physical table group does not contain any object identifiers.

Physical Table MIB objects are listed in Table 4-8.

Table 4-8 Physical Table MIB Objects

Name	Syntax	Description	Default Value
PhysicalIndex	`Integer (1..2147483647)`	Specifies an arbitrary value that uniquely identifies the physical entity. The value can be a small positive integer; index values for different physical entities are not necessarily contiguous. Status: current	none
PhysicalClass	`Integer {other(1), unknown(2), chassis(3), backplane(4), container(5), (for example, a chassis slot or daughter-card holder) powerSupply(6), fan(7), sensor(8), module(9), (for example, a plug-in card or daughter-card) port(10), stack(11) (for example, a stack of multiple chassis entities)}`	Specifies an enumerated value that provides an indication of the general hardware type of a particular physical entity. There are no restrictions as to the number of `entPhysicalEntries` of each `entPhysicalClass` that must be instantiated by an agent. The following are the definitions for enumeration: • `other`—Needs to know the physical entity class. It does not match any of the supported values. • `unknown`—Needs to know the physical entity class. It is unknown to the agent. • `chassis`—Needs to know if the physical entity class is an overall container for networking equipment. Any class of physical entity except a stack is contained within a chassis, and a chassis is contained only within a stack. • `backplane`—Needs to know if the physical entity class is a device for aggregating and forwarding networking traffic, such as a shared backplane in a modular Ethernet switch. An agent can model a backplane as a single physical entity, which is actually implemented as multiple discrete physical components (within a chassis or stack). • `container`—Needs to know if the physical entity class is capable of containing one or more removable physical entities, possibly of different types. For example, each (empty or full) slot in a chassis is modeled as a container. All removable physical entities are modeled within a container entity, such as field-replaceable modules, fans, or power supplies. All known containers are modeled by the agent, including empty containers.	none

Table 4-8 Physical Table MIB Objects (continued)

Name	Syntax	Description	Default Value
PhysicalClass (continued)		• powerSupply—Needs to know if the physical entity class is a power-supplying component. • fan—Needs to know if the physical entity class is a fan or other heat-reduction component. • sensor—Needs to know if the physical entity class is a sensor, such as a temperature sensor within a router chassis. • module—Needs to know if the physical entity class is a self-contained subsystem. If it is removable, it is modeled within a container entity; otherwise, it is modeled directly within another physical entity, for example, a chassis or another module. • port—Needs to know if the physical entity class is a networking port, capable of receiving or transmitting networking traffic. • stack—Needs to know if the physical entity class is a super-container (possibly virtual), intended to group together multiple, chassis entities. A stack is used by a virtual cable or a real interconnect cable. A stack can also be attached to multiple chassis or is comprised of multiple interconnect cables. A stack is not modeled within any other physical entities, but a stack is contained within another stack. Only chassis entities are contained within a stack. Status: current	
SnmpEngineId OrNone	Octet String (SIZE(0..32))—empty string or SnmpEngineID	Specifies a specially formatted SnmpEngineID string for use with the ENTITY MIB. If an instance of syntax object SnmpEngineIdOrNone has a nonzero length, the object encoding and semantics are defined by the SnmpEngineID textual convention (refer to RFC 2571). If an instance of syntax object SnmpEngineIdOrNone contains a zero-length string, no appropriate SnmpEngineID is associated with the logical entity (that is, SNMPv3 is not supported). Status: current	none

Physical Entity Table

The object identifiers are listed in Table 4-9.

Table 4-9 Physical Entity Table Object Identifiers

Name	Object Identifier
EntPhysicalTable	::= { entityPhysical 1 }
entPhysicalEntry	::= { entPhysicalTable 1 }
entPhysicalIndex	::= { entPhysicalEntry 1 }
entPhysicalDescr	::= { entPhysicalEntry 2 }
entPhysicalVendorType	::= { entPhysicalEntry 3 }
entPhysicalContainedIn	::= { entPhysicalEntry 4 }
entPhysicalClass	::= { entPhysicalEntry 5 }
entPhysicalParentRelPos	::= { entPhysicalEntry 6 }
entPhysicalName	::= { entPhysicalEntry 7 }
entPhysicalHardwareRev	::= { entPhysicalEntry 8 }
entPhysicalFirmwareRev	::= { entPhysicalEntry 9 }
entPhysicalSoftwareRev	::= { entPhysicalEntry 10 }
entPhysicalSerialNum	::= { entPhysicalEntry 11 }
entPhysicalMfgName	::= { entPhysicalEntry 12 }
entPhysicalModelName	::= { entPhysicalEntry 13 }
entPhysicalAlias	::= { entPhysicalEntry 14 }
entPhysicalAssetID	::= { entPhysicalEntry 15 }
entPhysicalIsFRU	::= { entPhysicalEntry 16 }

The MIB objects are described in Table 4-10.

Table 4-10 Physical Entity Table MIB Objects

Name	Syntax	Description	Default Value
EntPhysicalTable	`Sequence of EntPhysicalEntry`	Contains one row per physical entity. There is always at least one row for an overall physical entity. Max-Access: not-accessible Status: current	none
entPhysicalEntry	`entPhysicalEntry`	Provides information about a particular physical entity. Each entry provides objects `entPhysicalDescr`, `entPhysicalVendorType`, and `entPhysicalClass` to help an NMS identify and characterize the entry. The objects `entPhysicalContainedIn` and `entPhysicalParentRelPos` relate this entry to other entries in this table. For the OID[1] values assigned to the `entPhysicalVendorType`, see Table 1-8, Table 1-9, and Table 1-10. Max-Access: not-accessible Status: current The index contains `entPhysicalIndex`.	none
entPhysicalIndex	`PhysicalIndex`	Specifies the index for this entry. Max-Access: not-accessible Status: current	none
entPhysicalDescr	`SnmpAdminString`	Provides a textual description for a physical entity. This object contains a string that identifies the manufacturer name for the physical entity, and is set to a different value for each version or model of the physical entity. Max-Access: read-only Status: current	none

Table 4-10 Physical Entity Table MIB Objects (continued)

Name	Syntax	Description	Default Value
entPhysicalVendorType	AutonomousType	Indicates the vendor-specific hardware type of the physical entity. **Note** This is different from the definition of MIB-II sysObjectID. An agent sets this object to a enterprise-specific registration identifier value indicating the specific equipment type in detail. The associated instance of entPhysicalClass is used to indicate the general type of hardware device. If no vendor-specific registration identifier exists for this physical entity, or the value is unknown by this agent, the value { 0 0 } is returned. For OID values of this object, see Table 1-8, Table 1-9, and Table 1-10. Max-Access: read-only Status: current	none
entPhysicalContainedIn	Integer (0..2147483647)	Specifies the value of entPhysicalIndex for the physical entity containing this object. A value of 0 indicates this physical entity is not contained in any other physical entity. **Note** The set of containment relationships defines a strict hierarchy; that is, recursion is not allowed. In the event a physical entity is contained by more than one physical entity (for example, double-wide modules), this object can identify the containing entity with the lowest value of entPhysicalIndex. Max-Access: read-only Status: current	none
entPhysicalClass	PhysicalClass	Indicates the general hardware type of the physical entity. An agent can set this object to the standard enumeration value that most accurately indicates the general class of the physical entity, or the primary class if there is more than one. If no appropriate standard registration identifier exists for this physical entity, the value other(1) is returned. If the value is unknown by this agent, the value unknown(2) is returned. Max-Access: read-only Status: current	none

Table 4-10 Physical Entity Table MIB Objects (continued)

Name	Syntax	Description	Default Value
entPhysicalParentRelPos	`Integer (-1..2147483647)`	Indicates the relative position of this child component among all its sibling components. Sibling components are defined as `entPhysicalEntries`, which share the same instance values of each of the `entPhysicalContainedIn` and `entPhysicalClass` objects. An NMS can use this object to identify the relative ordering for all sibling components of a particular parent (identified by the `entPhysicalContainedIn` instance in each sibling entry). This value can match any external labeling of the physical component if possible. For example, for a container (card slot) labeled as slot #3, `entPhysicalParentRelPos` should have the value 3. **Note** The `entPhysicalEntry` for the module plugged in slot 3 can have an `entPhysicalParentRelPos` value of 1. If the physical position of this component does not match any external numbering or visible ordering, user documentation or other external reference material should be used to determine the parent-relative position. If this is not possible, the agent should assign a consistent (but possibly arbitrary) ordering to a given set of sibling components, based on internal representation of the components.	none

Table 4-10 Physical Entity Table MIB Objects (continued)

Name	Syntax	Description	Default Value
entPhysicalParentRelPos (continued)		If the agent cannot determine the parent-relative position, or if the associated value of `entPhysicalContainedIn` is 0, the value `-1` is returned. Otherwise, a nonnegative integer is returned, which indicates the parent-relative position of this physical entity. Parent-relative ordering normally starts from `1` and continues to *n*. For example, *n* represents the highest positioned child entity. If the physical entities, for example, slots, are labeled from a starting position of 0, the first sibling is associated with a `entPhysicalParentRelPos` value 0. **Note** This ordering can be sparse or dense, depending on agent implementation. The actual values returned are not globally meaningful, as each parent component can use different numbering algorithms. The ordering is meaningful only among siblings of the same parent component. The agent can retain parent-relative position values across reboots, either through algorithmic assignment or use of nonvolatile storage. Max-Access: read-only Status: current	
entPhysicalName	`SnmpAdminString`	Specifies the textual name of the physical entity. The value of this object can be the name of the component as assigned by the local device, and can be suitable for use in commands entered at the device console. This can be a text name, such as console, or a simple component number (for example, port or module number) such as 1, depending on the physical component naming syntax of the device. If there is no local name, or this object is not applicable, this object contains a zero-length string. **Note** The value of `entPhysicalName` for two physical entities can be the same if the console interface does not distinguish between them, for example, slot-1 and the card in slot-1. Max-Access: read-only Status: current	none

Table 4-10 Physical Entity Table MIB Objects (continued)

Name	Syntax	Description	Default Value
entPhysicalHardwareRev	SnmpAdminString	Specifies the vendor-specific hardware revision string for the physical entity. The preferred value is the hardware revision identifier printed on the component itself (if present). **Note** If revision information is stored internally in a nonprintable format, (for example, binary), the agent must convert the information to a printable format in an implementation-specific manner. If no specific hardware revision string is associated with the physical component, or this information is unknown to the agent, this object can contain a zero-length string. Max-Access: read-only Status: current	none
entPhysicalFirmwareRev	SnmpAdminString	Specifies the vendor-specific firmware revision string for the physical entity. **Note** If revision information is stored internally in a non-printable format (for example, binary), the agent must convert the information to a printable format in an implementation-specific manner. If no specific firmware programs are associated with the physical component, or this information is unknown to the agent, this object can contain a zero-length string. Max-Access: read-only Status: current	none
entPhysicalSoftwareRev	SnmpAdminString	Specifies the vendor-specific software revision string for the physical entity. **Note** If revision information is stored internally in a nonprintable format (for example, binary), the agent must convert the information to a printable format in an implementation-specific manner. If no specific software programs are associated with the physical component, or this information is unknown to the agent, this object can contain a zero-length string. Max-Access: read-only Status: current	none

Table 4-10 Physical Entity Table MIB Objects (continued)

Name	Syntax	Description	Default Value
entPhysicalSerialNum	`SnmpAdminString (SIZE (0..32))`	Specifies the vendor-specific serial number string for the physical entity. The preferred value is the serial number string printed on the component itself (if present). On the first instantiation of an physical entity, the associated value `entPhysicalSerialNum` is set to the correct vendor-assigned serial number, if this information is available to the agent. If a serial number is unknown or nonexistent, the `entPhysicalSerialNum` is set to a zero-length string instead. **Note** The implementations can identify the serial numbers of all installed physical entities, which do not provide write access to the `entPhysicalSerialNum` object. Agents which cannot provide nonvolatile storage for the `entPhysicalSerialNum` strings are not required to implement write access for this object. Not every physical component has a serial number, or needs one. Physical entities for which the associated value of the `entPhysicalIsFRU` object is equal to `false(2)` (for example, the repeater ports within a repeater module) do not need a unique serial number. An agent does not need to provide write access for such entities and can return a zero-length string. If write access is implemented for an instance of `entPhysicalSerialNum`, and a value is written into the instance, the agent must retain the supplied value in the `entPhysicalSerialNum` instance associated with the same physical entity for as long as that entity remains instantiated. This includes instantiations across all reinitializations and reboots of the network management system, including those which result in a change of the physical entity `entPhysicalIndex` value. Max-Access: read-write Status: current	none

Table 4-10 Physical Entity Table MIB Objects (continued)

Name	Syntax	Description	Default Value
entPhysicalMfgName	`SnmpAdminString`	Specifies the manufacturer name of the physical component. The preferred value is the manufacturer name string printed on the component itself (if present). **Note** The comparisons between instances of the `entPhysicalModelName`, `entPhysicalFirmwareRev`, `entPhysicalSoftwareRev`, and the `entPhysicalSerialNum` objects, are only meaningful among `entPhysicalEntries` with the same value of `entPhysicalMfgName`. If the manufacturer name string associated with the physical component is unknown to the agent, this object contains a zero-length string. Max-Access: read-only Status: current	none
entPhysicalModelName	`SnmpAdminString`	Specifies the model name identifier string associated with this physical component. It is different for each vendor. The preferred value is the manufacturer-visible part number, which is printed on the component itself. If the model name string associated with the physical component is unknown to the agent, this object contains a zero-length string. Max-Access: read-only Status: current	none

Table 4-10 Physical Entity Table MIB Objects (continued)

Name	Syntax	Description	Default Value
entPhysicalAlias	SnmpAdminString (SIZE (0..32))	Specifies an alias name for the physical entity as specified by a network manager, and provides a nonvolatile handle for the physical entity. On the first instantiation of a physical entity, the value of entPhysicalAlias is set to the zero-length string. However, the agent sets the value to a locally unique default value, instead of a zero-length string. If write access is implemented for an instance of entPhysicalAlias, and a value is written into the instance, the agent must retain the supplied value in the entPhysicalAlias instance associated with the same physical entity. This includes instantiations across all reinitializations and reboots of the network management system, including those which result in a change of the physical entity entPhysicalIndex value. Max-Access: read-write Status: current	none

Table 4-10 Physical Entity Table MIB Objects (continued)

Name	Syntax	Description	Default Value
entPhysicalAssetID	`SnmpAdminString (SIZE (0..32))`	Specifies a user-assigned asset tracking identifier for the physical entity as specified by a network manager, and provides nonvolatile storage of this information. On the first instantiation of a physical entity, the value `entPhysicalAssetID` is set to the zero-length string. Not every physical component will have an asset tracking identifier, or even need one. Physical entities where the associated value of the `entPhysicalIsFRU` object is equal to `false(2)` (that is, the repeater ports within a repeater module), do not a unique asset tracking identifier. An agent does not have to provide write access for such entities, and may instead return a zero-length string. If write access is implemented for an instance of `entPhysicalAssetID`, and a value is written into the instance, the agent must retain the supplied value in the `entPhysicalAssetID` instance associated with the same physical entity for as long as that entity remains instantiated. This includes instantiations across all reinitializations and reboots of the network management system, including those which result in a change of the physical entity `entPhysicalIndex` value. If no asset tracking information is associated with the physical component, this object contains a zero-length string. Max-Access: read-write Status: current	none
entPhysicalIsFRU	`TruthValue`	Indicates whether or not this physical entity is a FRU. If this object contains the value `true(1)`, this `entPhysicalEntry` identifies a field replaceable unit. For all `entPhysicalEntries` which represent components that are permanently contained within a field replaceable unit, the value `false(2)` is returned for this object. Max-Access: read-only Status: current	none

1. OID = object identifier

Physical Mapping Table

The object identifier for each MIB object is listed in Table 4-11.

Table 4-11 Physical Mapping Table Object Identifiers

Name	Object Identifier
entPhysicalContainsTable	::= { entityMapping 3 }
entPhysicalContainsEntry	::= { entPhysicalContainsTable 1 }
entPhysicalChildIndex	::= { entPhysicalContainsEntry 1 }

The MIB objects are listed in Table 4-12.

Table 4-12 Physical Mapping Table MIB Objects

Name	Syntax	Description	Default Value
entPhysicalContainsTable	`Sequence of entPhysicalContainsEntry`	Shows the container 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. If a physical entity is contained by more than one other physical entity (for example, double-wide modules), this table can include these additional mappings, which cannot be represented in the `entPhysicalTable` virtual containment tree. Max-Access: not-accessible Status: current	none
entPhysicalContainsEntry	`entPhysicalContainsEntry`	Indicates a single container or containee relationship. Max-Access: not-accessible Status: current The indexes are the following: • `entPhysicalIndex` • `entPhysicalChildIndex`	none
entPhysicalChildIndex	`PhysicalIndex`	Specifies the value `entPhysicalIndex` for the contained physical entity. Max-Access: read-only Status: current	none

Last Change TimeStamp

The object identifier for the MIB object is shown in Table 4-13.

Table 4-13 Last Change TimeStamp Object Identifier

Name	Object Identifier
entLastChangeTime	::= { entityGeneral 1 }

The MIB object is listed in Table 4-14.

Table 4-14 Last Change TimeStamp MIB Object

Name	Syntax	Description	Default Value
entLastChangeTime	`TimeStamp`	Specifies the value of `sysUpTime` when a conceptual row is created, modified, or deleted in these tables: • `entPhysicalTable` • `entLogicalTable` • `entLPMappingTable` • `entAliasMappingTable` • `entPhysicalContainsTable` Max-Access: read-only Status: current	none

Entity Conformance and Compliance Information

The object identifiers are listed in Table 4-15.

Table 4-15 Entity Conformance Groups

Name	Object Identifier
entityConformance	::= { entityMIB 3 }
entityCompliances	::= { entityConformance 1 }
entityGroups	::= { entityConformance 2 }

Entity Compliance Statements

One compliance object identifier is listed in Table 4-16.

Table 4-16 Entity Compliance Object Identifier

Name	Object Identifier
entPhysicalAssetID	::= { entityCompliances 2 }

A list of the objects for compliance is listed in Table 4-17.

Table 4-17 Entity Compliance Objects

Name	Mandatory Groups	Description	Default Value
entity2Compliance	`entityPhysicalGroup`, `entityPhysical2Group`, `entityGeneralGroup` `entityNotificationsGroup`	Describes the compliance statement for SNMP entities that implement version 2 of the entity MIB. Status: current	none
entityLogical2Group	none	Specifies the implementation of this group is not mandatory for agents which model all MIB object instances within a single naming scope.	none
entityMappingGroup	none	Implementation of the `entPhysicalContainsTable` is mandatory for all agents. Implementation of the `entLPMappingTable` and `entAliasMappingTables` are not mandatory for agents which model all MIB object instances within a single naming scope. **Note** The `entAliasMappingTable` can be useful for all agents; however, implementation of the `entityLogicalGroup` or `entityLogical2Group` is required to support this table.	none
entPhysicalSerialNum	none	Specifies read and write access are not required for agents that cannot identify serial number information for physical entities, and cannot provide non-volatile storage for NMS-assigned serial numbers. Write access is not required for agents that can identify serial number information for physical entities, but cannot provide non-volatile storage for NMS-assigned serial numbers. Write access is not required for physical entities for physical entities for which the associated value of the `entPhysicalIsFRU` object is equal to `false(2)`. Min-Access: not-accessible	none
entPhysicalAlias	none	Specifies write access is required only if the associated `entPhysicalClass` value is equal to `chassis(3)`. Min-Access: read-only	none
entPhysicalAssetID	none	Specifies read and write access is not required for agents that cannot provide non-volatile storage for NMS-assigned asset identifiers. Write access is not required for physical entities for which the associated value of `entPhysicalIsFRU` is equal to `false(2)`. Min-Access: not-accessible	none

Entity MIB Groupings

The object identifiers are listed in Table 4-18.

Table 4-18 Entity MIB Groupings Object Identifiers

Name	Object Identifier
entityPhysicalGroup	::= { entityGroups 1 }
entityMappingGroup	::= { entityGroups 3 }
entityGeneralGroup	::= { entityGroups 4 }
entityNotificationsGroup	::= { entityGroups 5 }
entityPhysical2Group	::= { entityGroups 6 }
entityLogical2Group	::= { entityGroups 7 }

The object groups are listed in Table 4-19.

Table 4-19 Entity MIB Groupings Objects

Name	Objects	Description	Default Value
entityPhysicalGroup	`entPhysicalDescr`, `entPhysicalVendorType`, `entPhysicalContainedIn`, `entPhysicalClass`, `entPhysicalParentRelPos`, `entPhysicalName`	Specifies the collection of objects used to represent physical system components, which provide management information for a single agent. Status: current	none
entityMappingGroup	`entLPPhysicalIndex`, `entAliasMappingIdentifier`, `entPhysicalChildIndex`	Specifies the collection of objects used to represent the associations between multiple logical entities, physical components, interfaces, and port identifiers. A single agent provides management information for the applicable objects. Status: current	none
entityGeneralGroup	`entLastChangeTime`	Specifies the collection of objects used to represent general entity information. A single agent provides management information. Status: current	none
entityNotificationsGroup	`entConfigChange`	Specifies the collection of notifications used to indicate consistent Entity MIB data and general status information. **Note** The `entityNotificationsGroup` object applicable to the notification group. `entConfigChange` is a notification object. Status: current	none

Table 4-19 Entity MIB Groupings Objects (continued)

Name	Objects	Description	Default Value
entityPhysical2Group	`entPhysicalHardwareRev`, `entPhysicalFirmwareRev`, `entPhysicalSoftwareRev`, `entPhysicalSerialNum`, `entPhysicalMfgName`, `entPhysicalModelName`, `entPhysicalAlias`, `entPhysicalAssetID`, `entPhysicalIsFRU`	Specifies the collection of objects used to represent physical system components. A single agent provides management information. This group augments the objects contained in the `entityPhysicalGroup`. Status: current	none
entityLogical2Group	`entLogicalDescr`, `entLogicalType`, `entLogicalTAddress`, `entLogicalTDomain`, `entLogicalContextEngineID`, `entLogicalContextName`	Specifies the collection of objects used to represent the list of logical entities. A single SNMP entity provides management information. Status: current	none

Sensor MIB Objects

This section describes the individual MIB objects that make up the PXM45-specific MIB files. The SENSOR MIB resides in the CISCO-ENTITY-SENSOR-MIB.my file. It is used to monitor the values of sensors in the ENTITY MIB (refer to RFC 2037) `entPhysicalTable`.

The Sensor MIB objects include:

- SENSOR MIB Textual Conventions
- SENSOR MIB Objects Table
- Ent Sensor Threshold Table
- SENSOR MIB Notification
- SENSOR MIB Conformance Information

SENSOR MIB Textual Conventions

The names of the textual conventions are specified in the object syntax.

The MIB objects are listed in Table 4-20.

Note Textual Conventions do not have any object identifiers.

Table 4-20 SENSOR MIB Textual Conventions MIB Objects

Name	Syntax	Description	Default Value
SensorDataType	`Integer {other (1), unknown (2), voltsAC (3), voltsDC (4), amperes (5), watts (6), hertz (7), celsius (8), percentRH (9), rpm (10), cmm (11), truthvalue (12), specialEnum (13)}`	Describes the valid values that are used for `SensorDataType`. Status: current The following are the values for `SensorDataType`:	none

Sensor Measurement	Valid Value
`other(1)`	Measure other than those listed below
`unknown(2)`	Unknown measurement, or arbitrary, relative numbers
`voltsAC(3)`	Electric potential
`voltsDC(4)`	Electric potential
`amperes(5)`	Electric current
`watts(6)`	Power
`hertz(7)`	Frequency
`celsius(8)`	Temperature
`percentRH(9)`	Percent relative humidity
`rpm(10)`	Shaft revolutions per minute
`cmm(11)`	Cubic meters per minute (airflow)
`truthvalue(12)`	Value takes `{true(1), false(2)}`
`specialEnum(13)`	Value takes user-defined enumerated values

Table 4-20 SENSOR MIB Textual Conventions MIB Objects (continued)

Name	Syntax	Description	Default Value
SensorDataScale	`Integer {yocto (1), zepto (2), atto (3), femto (4), pico (5), nano (6), micro (7), milli (8), Units (9), kilo (10), mega (11), giga (12), tera (13), exa (14), peta (15), zetta (16), yotta (17)}`	Displays the International System of Units prefixes. Status: current The following list the prefix and the value:	none

Prefix	Value
`yocto (1)`	10^{-24}
`zepto (2)`	10^{-21}
`atto (3)`	10^{-18}
`femto (4)`	10^{-15}
`pico (5)`	10^{-12}
`nano (6)`	10^{-9}
`micro (7)`	10^{-6}
`milli (8)`	10^{-3}
`Units (9)`	10^{0}
`kilo (10)`	10^{3}
`mega (11)`	10^{6}
`giga (12)`	10^{9}
`tera (13)`	10^{12}
`exa (14)`	10^{15}
`peta (15)`	10^{18}
`zetta (16)`	10^{21}
`yotta (17)`	10^{24}

Table 4-20 SENSOR MIB Textual Conventions MIB Objects (continued)

Name	Syntax	Description	Default Value
SensorPrecision	`Integer (-8..9)`	Determines the number of decimal places in the fractional part of a `SensorValue` fixed-point number. When the range is from -8 to -1, `SensorPrecision` is the number of accurate digits in a `SensorValue` fixed-point number. `SensorPrecision` is 0 for nonfixed-point numbers. Agents must choose a value for `SensorPrecision` so that the precision and accuracy of a `SensorValue` is correctly indicated. For example, a temperature sensor that measures from 0° to 100° C in 0.1° increments of ±0.05° would have a `SensorPrecision` of 1, a `SensorDataScale` of Units (0), and a `SensorValue` ranging from 0 to 1000. The `SensorValue` is interpreted as (°C * 10). If that temperature sensor precision were 0.1° but the accuracy is only ±0.5°, the `SensorPrecision` would be set to 0. The `SensorValue` is interpreted as degrees C. Another example, a fan rotation speed sensor that measures RPM from 0 to 10,000 in 100 RPM increments, with an accuracy of +50/-37 RPM, can have a `SensorPrecision` of -2, a `SensorDataScale` of `Units(9)`, and a `SensorValue` ranging from 0 to 10000. The 10s and 1s digits of `SensorValue` is always 0. Status: current	none

Table 4-20 SENSOR MIB Textual Conventions MIB Objects (continued)

<table>
<tr><th>Name</th><th>Syntax</th><th>Description</th><th>Default Value</th></tr>
<tr><td>SensorValue</td><td>Integer
(-1000000000..1000000000)</td><td>Measures AC volts, DC volts, amperes, watts, hertz, Celsius, and cmm. SensorValue is a fixed point number ranging from -999,999,999 to +999,999,999. Use the value -1000000000 to indicate underflow. Use the value +1000000000 to indicate overflow. Use SensorPrecision to indicate how many fractional digits the SensorValue has.

For sensors that measure percent RH, this item is a number ranging from 0 to 100.

For sensors that measure RPM, this item can take only nonnegative values, 0..999999999.

For sensors of type truthvalue, this item can take only two values: true(1) and false(2).

For sensors of type specialEnum, this item can take any value in the range (-1000000000..1000000000), but the meaning of each value is specific to the sensor.

For sensors of type other and unknown, this item can take any value in the range (-1000000000..1000000000), but the meaning of the values are specific to the sensor.

Use ENTITY MIB entPhysicalTable and entPhysicalVendorType to learn about the sensor type. For OID[1] values assigned to entPhysicalVendorType, see Table 1-8, Table 1-9, and Table 1-10.

Status: current</td><td>none</td></tr>
<tr><td>SensorStatus</td><td>Integer {ok (1),
unavailable (2),
nonoperational (3)}</td><td>Indicates the operational status of the sensor.

Status: current

The following are the values for SensorStatus:

<table><tr><th>Sensor Measurement</th><th>Valid Value</th></tr><tr><td>ok (1)</td><td>The agent can read the sensor value.</td></tr><tr><td>unavailable (2)</td><td>The agent presently cannot report the sensor value.</td></tr><tr><td>nonoperational (3)</td><td>The sensor can have a hard failure (disconnected wire), or a soft failure such as out-of-range, jittery, or fluctuating readings.</td></tr></table></td><td>none</td></tr>
</table>

Table 4-20 SENSOR MIB Textual Conventions MIB Objects (continued)

Name	Syntax	Description	Default Value
SensorValue UpdateRate	`Integer (0..999999999)`	Indicates the interval in seconds between updates to the sensor value. A value of 0 indicates one or more of the following: • Sensor value is updated on demand (when polled by the agent for a get-request). • Sensor value changes (event-driven). Agent does not know the rate. Status: current	none
SensorThreshold Severity	`Integer {other (1), minor (10), major (20)}`	Indicates the sensor threshold severity. Status: current The following are the values for `SensorThresholdSeverity`:	none

Sensor Measurement	Valid Value
`other (1)`	Indicates a severity other than those listed below.
`minor (10)`	Minor problem threshold.
`major (20)`	Major problem threshold.

Name	Syntax	Description	Default Value
SensorThreshold Relation	`Integer {lessThan (1), lessOrEqual (2), greaterThan (3), greaterOrEqual (4), equalTo (5), notEqualTo (6)}`	Displays the sensor threshold relational operator types. Status: current The following are the values for `SensorThresholdRelation`:	none

Sensor Threshold	Valid Value
`lessThan(1)`	Sensor value is less than the threshold value.
`lessOrEqual(2)`	Sensor value is less than or equal to the threshold value.
`greaterThan(3)`	Sensor value is greater than the threshold value.
`greaterOrEqual(4)`	Sensor value is greater than or equal to the threshold value.
`equalTo(5)`	Sensor value is equal to the threshold value.
`notEqualTo(6)`	Sensor value is not equal to the threshold value.

1. OID = object identifier

SENSOR MIB Objects Table

The object identifier for each MIB object is listed in Table 4-21.

Table 4-21 SENSOR MIB Object Identifiers

Name	Object Identifier
entSensorValueTable	::= { entSensorValues 1 }
entSensorValueEntry	::= { entSensorValueTable 1 }
entSensorType	::= { entSensorValueEntry 1 }
entSensorScale	::= { entSensorValueEntry 2 }
entSensorPrecision	::= { entSensorValueEntry 3 }
entSensorValue	::= { entSensorValueEntry 4 }
entSensorStatus	::= { entSensorValueEntry 5 }
entSensorValueTimeStamp	::= { entSensorValueEntry 6 }
entSensorValueUpdateRate	::= { entSensorValueEntry 7 }

The MIB objects are described in Table 4-22.

Table 4-22 SENSOR MIB Objects Table

Name	Syntax	Description	Default Value
entSensorValueTable	`Sequence of entSensorValueEntry`	Lists the type, scale, and present value of a sensor listed in the ENTITY MIB `entPhysicalTable`. Max-Access: not-accessible Status: current	none
entSensorValueEntry	`entSensorValueEntry`	Describes the present reading of a sensor, the measurement units and scale, and the sensor operational status. Max-Access: not-accessible Status: current The index contains `entPhysicalIndex`.	none
entSensorType	`SensorDataType`	Indicates the type of data reported by the `entSensorValue`. This object is set by the agent at startup and the value does not change during operation. Max-Access: read-only Status: current	none
entSensorScale	`SensorDataScale`	Indicates the exponent to apply to sensor values reported by `entSensorValue`. This object is set by the agent at start-up and the value does not change during operation. Max-Access: read-only Status: current	none

Table 4-22 SENSOR MIB Objects Table (continued)

Name	Syntax	Description	Default Value
entSensorPrecision	SensorPrecision	Indicates the number of decimal places of precision in fixed-point sensor values reported by entSensorValue. This object is set to 0 when entSensorType is not a fixed-point type voltsAC (1), voltsDC (2), amperes (3), watts (4), hertz (5), celsius (6), or cmm (9). This object is set by the agent at start-up and the value does not change during operation. Max-Access: read-only Status: current	none
entSensorValue	SensorValue	Reports the most recent measurement seen by the sensor. To correctly display or interpret this object value, you must also know entSensorType, entSensorScale, and entSensorPrecision. However, you can compare entSensorValue with the threshold values given in entSensorThresholdTable without any semantic knowledge. Max-Access: read-only Status: current	none
entSensorStatus	SensorStatus	Indicates the present operational status of the sensor. Max-Access: read-only Status: current	none
entSensorValueTimeStamp	TimeStamp	Indicates the age of the value reported by entSensorValue. Max-Access: read-only Status: current	none
entSensorValueUpdateRate	SensorValueUpdateRate	Indicates the rate that the agent updates entSensorValue. Units: seconds Max-Access: read-only Status: current	none

Ent Sensor Threshold Table

The object identifier for each MIB object is listed in Table 4-23.

Table 4-23 Ent Sensor Threshold Table Object Identifiers

Name	Object Identifier
entSensorThresholdTable	::= { entSensorThresholds 1 }
entSensorThresholdEntry	::= { entSensorThresholdTable 1 }
entSensorThresholdIndex	::= { entSensorThresholdEntry 1 }
entSensorThresholdSeverity	::= { entSensorThresholdEntry 2 }
entSensorThresholdRelation	::= { entSensorThresholdEntry 3 }
entSensorThresholdValue	::= { entSensorThresholdEntry 4 }
entSensorThresholdEvaluation	::= { entSensorThresholdEntry 5 }
entSensorThresholdNotificationEnable	::= { entSensorThresholdEntry 6 }

The MIB objects are described in Table 4-24.

Table 4-24 Ent Sensor Threshold Table MIB Objects

Name	Syntax	Description	Default Value
entSensorThreshold Table	`Sequence of entSensorThresholdEntry`	Lists the threshold severity, relation, and comparison value for a sensor listed in the ENTITY MIB `entPhysicalTable`. Max-Access: not-accessible Status: current	none
entSensorThreshold Entry	`entSensorThresholdEntry`	Describes the thresholds for a sensor, the threshold severity, the threshold value, the relation, and the evaluation of the threshold. Only entities of type `sensor (8)` are listed in this table. Only preconfigured thresholds are listed in this table. Users can create sensor-value monitoring instruments in different ways, such as Remote Network Monitoring (RMON) alarm, or Expression-MIB. Entries are created by the agent at system startup and FRU insertion. Entries are deleted by the agent at FRU removal. Max-Access: not-accessible Status: current The following are the indexes: • `entPhysicalIndex` • `entSensorThresholdIndex`	none

Table 4-24 Ent Sensor Threshold Table MIB Objects (continued)

Name	Syntax	Description	Default Value
entSensorThreshold Index	`Integer32 (1..99999999)`	Specifies an index that uniquely identifies an entry in the `entSensorThreshold` table. This index permits the same sensor to have several different thresholds. Max-Access: not-accessible Status: current	none
entSensorThreshold Severity	`SensorThresholdSeverity`	Indicates the severity of the sensor threshold. Max-Access: read-only Status: current	none

entSensorThreshold Relation	`SensorThresholdRelation`	Indicates the relation between sensor value (`entSensorValue`) and threshold value (`entSensorThresholdValue`), which are required to trigger the alarm when evaluating the relation. `entSensorValue` is on the left of `entSensorThresholdRelation` and `entSensorThresholdValue` is on the right.
		In pseudo-code, the evaluation-alarm mechanism is the following:
		`...`
		`if (entSensorStatus == ok) then`
		`if (evaluate(entSensorValue,`
		`entSensorThresholdRelation,`
		`entSensorThresholdValue))`
		`then`
		`if (entSensorThresholdNotificationEnable == true)) then`
		`raise_alarm(entSensorThresholdAlarmOID);`
		`endif`
		`endif`
		`endif`
		`...`
		Max-Access: read-only
		Status: current
			none

entSensorThreshold Value	`SensorValue`	Indicates the value of the threshold. To correctly display or interpret this object value, you must also know `entSensorType`, `entSensorScale`, and `entSensorPrecision`. However, you can directly compare `entSensorValue` with the threshold values given in `entSensorThresholdTable` without any semantic knowledge. Max-Access: read-only Status: current	none

Table 4-24 Ent Sensor Threshold Table MIB Objects (continued)

Name	Syntax	Description	Default Value
entSensorThreshold Evaluation	TruthValue	Indicates the result of the most recent evaluation of the threshold. If the threshold condition is true, entSensorThresholdEvaluation is true(1). If the threshold condition is false, entSensorThresholdEvaluation is false(2). Thresholds are evaluated at the rate indicated by entSensorValueUpdateRate. Max-Access: read-only Status: current	none
entSensorThreshold NotificationEnable	TruthValue	Controls the generation of entSensorThresholdNotification for this threshold. When this object is true (1), the generation of entSensorThresholdNotification is enabled. When this object is false(2), the generation of entSensorThresholdNotification is disabled. This object controls only the generation of entSensorThresholdNotification. Max-Access: read-write Status: current	none

SENSOR MIB Notification

According to RFC 2578, a notification is used to describe unsolicited transmissions of management information. NOTIFICATION-TYPE conveys the syntax and semantics of a notification.

The object identifier for the MIB object is listed in Table 4-25.

Table 4-25 SENSOR MIB Notification Object Identifier

Name	Object Identifier
entSensorThresholdNotification	::= { entitySensorMIBNotifications 1 }

The MIB object is listed in Table 4-26.

Table 4-26 SENSOR MIB Notification MIB Object

Name	Objects	Description	Default Value
entSensorThreshold Notification	entSensorThresholdValue, entSensorValue	Generates a notification each time the sensor value crosses the threshold listed in entSensorThresholdTable. The agent implementation facilitates a prompt evaluation of threshold and generation. Status: current	none

SENSOR MIB Conformance Information

The object identifier used for the SENSOR MIB conformance is listed in Table 4-27.

Table 4-27 SENSOR MIB Conformance Group

Name	Object Identifier
entitySensorMIBComplianceV01	::= { entitySensorMIBCompliances 1 }

Table 4-28 lists one object for conformance.

Table 4-28 SENSOR MIB Conformance Object

Name	Module	Description	Default Value
entitySensorMIB ComplianceV01	PXM45	Lists sensors in the `entPhysicalTable`. The `entitySensorMIBComplianceV01` object must implement this group as an ENTITY MIB. Status: current The mandatory groups are: • `entitySensorValueGroup` • `entitySensorThresholdGroup` • `entitySensorThresholdNotificationGroup`	none

SENSOR MIB Units of Conformance

The object identifier is listed in Table 4-29 for units of conformance.

Table 4-29 SENSOR MIB Units of Conformance

Name	Object Identifier
entitySensorValueGroup	::= { entitySensorMIBGroups 1 }
entitySensorThresholdGroup	::= { entitySensorMIBGroups 2 }
entitySensorThresholdNotificationGroup	::= { entitySensorMIBGroups 3 }

Table 4-30 lists the MIB objects for units of conformance.

Table 4-30 SENSOR MIB Units of Conformance Objects

Name	Objects	Description	Default Value
entitySensorValue Group	`entSensorType, entSensorScale, entSensorPrecision, entSensorValue, entSensorStatus, entSensorValueTimeStamp, entSensorValueUpdateRate`	Describes and monitors the values of the ENTITY MIB `entPhysicalTable` entries of sensors. Status: current	none
entitySensorThreshold Group	`entSensorThresholdSeverity, entSensorThresholdRelation, entSensorThresholdValue, entSensorThresholdEvaluation, entSensorThresholdNotification Enable`	Describes and monitors the thresholds for sensors. Status: current	none
entitySensorThreshold NotificationGroup	none	Monitors the sensor threshold activity for this collection of notifications. **Note** This notification is commented out because SMIC does not recognize the Notification Group. Notifications: `entSensorThresholdNotification` Status: current	none

SNMPv2 MIB Objects

This section describes the individual MIB objects that comprise SNMPv2. The objects reside in the SNMPv2-MIB.my file.

The SNMPv2 MIB objects include:

- System Group
- Object Resource Information
- SNMP Group
- Information for Notification
- Well-Known Traps
- Set Group
- SNMPv2 Conformance and Compliance Statements
- SNMPv2 Textual Conventions

System Group

The system group comprises a collection of objects that are common to all managed systems.

The following is the main object identifier for the `snmpMIBObjects`:

```
::= { mib-2 1 }
```

The object identifier for each MIB object is listed in Table 4-31. For possible values for the system group, see Table 1-11.

Table 4-31 System Group Object Identifiers

Name	Object Identifier
sysDescr	::= { system 1 }
sysObjectID	::= { system 2 }
sysUpTime	::= { system 3 }
sysContact	::= { system 4 }
sysName	::= { system 5 }
sysLocation	::= { system 6 }
sysServices	::= { system 7 }

The MIB objects are listed in Table 4-32.

Table 4-32 System Group MIB Objects

Name	Syntax	Description	Default Value
sysDescr	`DisplayString (SIZE (0..255))`	Defines the textual description of the entity. This value includes a the full name and version identification of the system hardware type, software operating system, and networking software. Max-Access: read-only Status: current	none
sysObjectID	`OBJECT IDENTIFIER`	Specifies the vendor authoritative identification of the network management subsystem that is contained in the entity. This value is allocated within the Structure of Management Information (SMI) enterprise subtree `1.3.6.1.4.1`. It provides an easy and unambiguous means for determining the type of box being managed. For example, if vendor Flintstones, Inc., is assigned the subtree `1.3.6.1.4.1.4242`, it can assign the identifier `1.3.6.1.4.1.4242.1.1` to Fred Router. For the possible OID values assigned to `sysObjectID`, see Table 1-11. Max-Access: read-only Status: current	none
sysUpTime	`TimeTicks`	Determines the time (in hundredths of a second) since the network management portion of the system is last reinitialized. Max-Access: read-only Status: current	none
sysContact	`DisplayString (SIZE (0..255))`	Specifies the textual identification of the contact person for this managed node along with information on how to contact this person. If no contact information is known, the value is the zero-length string. Max-Access: read-write Status: current	none
sysName	`DisplayString (SIZE (0..255))`	Determines an administratively-assigned name for this managed node. By convention, this is the node fully-qualified domain name. If the name is unknown, the value is the zero-length string. Max-Access: read-write Status: current	none

Table 4-32 System Group MIB Objects (continued)

<table>
<tr><th>Name</th><th>Syntax</th><th>Description</th><th>Default Value</th></tr>
<tr><td>sysLocation</td><td>DisplayString (SIZE (0..255))</td><td>Determines the physical location of this node (for example, telephone closet, third floor). If the location is unknown, the value is the zero-length string.
Max-Access: read-write
Status: current</td><td>none</td></tr>
<tr><td>sysServices</td><td>INTEGER (0..127)</td><td>Indicates the value for the set of services that this entity can potentially offer. The value is a sum. This sum initially takes the value 0. For each layer, L, in the range 1 to 7, this node performs transactions for 2 raised to (L - 1) is added to the sum. For example, a node that performs only routing functions can have a value of 4 (2^(3-1)). In contrast, a node which is a host offering application services can have a value of 72 (2^(4-1) + 2^(7-1)).
Max-Access: read-only
Status: current
For systems including Open System Interconnection (OSI) protocols, layers 5 and 6 can also be counted.
Note The context of the Internet suite of protocols contains values that are calculated accordingly.
The following are the supported layers:
<table>
<tr><th>Layer</th><th>Function</th></tr>
<tr><td>1</td><td>Physical—Supports repeaters.</td></tr>
<tr><td>2</td><td>Datalink/Subnetwork—Supports bridges.</td></tr>
<tr><td>3</td><td>Internet—Supports Internet Protocol (IP).</td></tr>
<tr><td>4</td><td>End-To-End—Supports Transmission Control Protocol (TCP).</td></tr>
<tr><td>7</td><td>Application—Supports Simple Management Transfer Protocol (SMTP).</td></tr>
</table></td><td>none</td></tr>
</table>

Object Resource Information

The object resource information comprises a collection of objects, which describe the SNMPv2 entity either statically or dynamically configured to support various MIB modules.

The object identifier for each MIB object is shown in Table 4-33. For the values that are used for the object resource information, see Table 1-11.

Table 4-33 Object Resource Information Object Identifiers

Name	Object Identifier
sysORLastChange	::= { system 8 }
sysORTable	::= { system 9 }
sysOREntry	::= { sysORTable 1 }
sysORIndex	::= { sysOREntry 1 }
sysORID	::= { sysOREntry 2 }
sysORDescr	::= { sysOREntry 3 }
sysORUpTime	::= { sysOREntry 4 }

The MIB objects are listed in Table 4-34.

Table 4-34 Object Resource Information MIB Objects

Name	Syntax	Description	Default Value
sysORLastChange	`TimeStamp`	Specifies the value of `sysUpTime` at the time of the most recent change in state or value of any instance of `sysORID`. Max-Access: read-only Status: current	none
sysORTable	`SEQUENCE OF SysOREntry`	Lists the capabilities of the local SNMPv2 entity that acts as an agent role with respect to various MIB modules. SNMPv2 entities, which are dynamically configured to support MIB modules, can have a dynamically-varying number of conceptual rows. Max-Access: not-accessible Status: current	none
sysOREntry	`SysOREntry`	Contains an entry (conceptual row) in the `sysORTable`. Max-Access: not-accessible Status: current The index contains `sysORIndex`.	none
sysORIndex	`INTEGER (1..2147483647)`	Specifies the auxiliary variable that is used to identify instances of the columnar objects in the `sysORTable`. Max-Access: not-accessible Status: current	none

Table 4-34 Object Resource Information MIB Objects (continued)

Name	Syntax	Description	Default Value
sysORID	OBJECT IDENTIFIER	Specifies an authoritative identification of a capabilities statement with respect to various MIB modules. They are supported by the local SNMPv2 entity that acts as an agent role. Max-Access: read-only Status: current	none
sysORDescr	DisplayString	Defines a textual description of the capabilities identified by the corresponding instance of sysORID. Max-Access: read-only Status: current	none
sysORUpTime	TimeStamp	Specifies the value sysUpTime at the time this conceptual row is last instantiated. Max-Access: read-only Status: current	none

SNMP Group

The SNMP group comprises a collection of objects that provide the basic instrumentation and control of an SNMP entity.

The following is the main object identifier for snmp:

```
::= { mib-2 11 }
```

The object identifier for each MIB object is listed in Table 4-35. For values that are used for the SNMP group, see Table 1-11.

Table 4-35 SNMP Group Object Identifiers

Name	Object Identifier
snmpInPkts	::= { snmp 1 }
snmpInBadVersions	::= { snmp 3 }
snmpInBadCommunityNames	::= { snmp 4 }
snmpInBadCommunityUses	::= { snmp 5 }
snmpInASNParseErrs	::= { snmp 6 }
snmpEnableAuthenTraps	::= { snmp 30 }
snmpSilentDrops	::= { snmp 31 }
snmpProxyDrops	::= { snmp 32 }

The SNMP Group MIB objects are listed in Table 4-36.

Table 4-36 SNMP Group MIB Objects

Name	Syntax	Description	Default Value
snmpInPkts	Counter32	Specifies the total number of messages delivered to the SNMP entity from the transport service. Max-Access: read-only Status: current	none
snmpInBadVersions	Counter32	Determines the total number of SNMP messages delivered to the SNMP entity and used for an unsupported SNMP version. Max-Access: read-only Status: current	none
snmpInBadCommunityNames	Counter32	Determines the total number of SNMP messages delivered to the SNMP entity that used a SNMP community name not known to the entity. Max-Access: read-only Status: current	none
snmpInBadCommunityUses	Counter32	Determines the total number of SNMP messages delivered to the SNMP entity that represented a SNMP operation. This operation is not allowed by the SNMP community named in the message. Max-Access: read-only Status: current	none
snmpInASNParseErrs	Counter32	Determines the total number of ASN.1 or Basic Encoding rules (BER) errors encountered by the SNMP entity when decoding received SNMP messages. Max-Access: read-only Status: current	none
snmpEnableAuthenTraps	INTEGER { enabled(1), disabled(2) }	Indicates whether the SNMP entity is permitted to generate authenticationFailure traps. The value of this object overrides any configuration information. It provides a means whereby all authenticationFailure traps are disabled. **Note** It is strongly recommended that this object be stored in nonvolatile memory, so that it remains constant across reinitializations of the NMS. Max-Access: read-write Status: current	none

Table 4-36 SNMP Group MIB Objects (continued)

Name	Syntax	Description	Default Value
snmpSilentDrops	Counter32	Determines the total number of GetRequest-PDU, GetNextRequest-PDU, GetBulkRequest-PDU, SetRequest-PDU, and InformRequest-PDU delivered to the SNMP entity. The snmpSilentDrops object is silently dropped because the size of a reply that contains an alternate Response-PDU with an empty variable-bindings field, which is greater than either a local constraint or the maximum message size associated with the originator of the request. Max-Access: read-only Status: current	none
snmpProxyDrops	Counter32	Determines the total number of GetRequest-PDU, GetNextRequest-PDU, GetBulkRequest-PDU, SetRequest-PDU, and InformRequest-PDU delivered to the SNMP entity. The snmpProxyDrops object is silently dropped because the transmission of the (possibly translated) message to a proxy target failed in a manner (other than a time-out) that no Response-PDU is returned. Max-Access: read-only Status: current	none

Information for Notification

The notification information comprises a collection of objects which allow the SNMPv2 entity, when acting as an agent role, to be configured to generate the SNMPv2-Trap-PDU.

The following is the main object identifier for the snmpTrap:

```
::= { snmpMIBObjects 4 }
```

The object identifier for each MIB object is listed in Table 4-37. For values that are used for the information for notification, see Table 1-11.

Table 4-37 Information for Notification Object Identifiers

Name	Object Identifier
snmpTrapOID	::= { snmpTrap 1 }
snmpTrapEnterprise	::= { snmpTrap 3 }

The MIB objects are listed in Table 4-38.

Table 4-38 Information for Notification MIB Objects

Name	Syntax	Description	Default Value
snmpTrapOID	OBJECT IDENTIFIER	Specifies the authoritative identification of the notification currently being sent. This variable occurs as the second varbind in every SNMPv2-Trap-PDU and InformRequest-PDU. Max-Access: accessible-for-notify Status: current	none
snmpTrapEnterprise	OBJECT IDENTIFIER	Specifies the authoritative identification of the enterprise associated with the trap currently being sent. When a SNMPv2 proxy agent is mapping an RFC1157Trap-PDU into a SNMPv2-Trap-PDU, this variable occurs as the last varbind. Max-Access: accessible-for-notify Status: current	none

Well-Known Traps

The following is the main object identifier for the snmpTraps:

```
::= { snmpMIBObjects 5 }
```

Note Both **linkDown** (::= { snmpTraps 3 })and **linkUp** (::= { snmpTraps 4 }) traps are defined in RFC 1573. RFC 1213 defines **egpNeighborLoss** (::= { snmpTraps 6 }).

The object identifier for each MIB object is listed in Table 4-39. For possible values for the traps, see Table 1-11.

Table 4-39 Well-Known Traps Object Identifiers

Name	Object Identifier
coldStart	::= { snmpTraps 1 }
warmStart	::= { snmpTraps 2 }
authenticationFailure	::= { snmpTraps 5 }

The MIB objects are listed in Table 4-40.

Table 4-40 Well-Known Traps MIB Objects

Name	Syntax	Description	Default Value
coldStart	none	Signifies that the SNMPv2 entity acts as an agent role. This trap reinitializes itself so the configuration is altered. Status: current	none
warmStart	none	Signifies that the SNMPv2 entity acts as an agent role This trap reintializes itself so the configuration is unaltered. Status: current	none
authenticationFailure	none	Signifies that the SNMPv2 entity acts as an agent role. This trap receives a protocol message that is not properly authenticated. While all implementations of the SNMPv2 must be capable of generating this trap, the `snmpEnableAuthenTraps` object indicates whether this trap is generated. Status: current	none

Set Group

The set group comprises a collection of objects which allow several cooperating SNMPv2 entities, all acting as a manager role, to coordinate their use of the SNMPv2 set operation.

The following is the main object identifier for `snmpSet`:

```
::= { snmpMIBObjects 6 }
```

The object identifier for the MIB object is listed in Table 4-41. For values that are used for the set group, see Table 1-11.

Table 4-41 Set Group Object Identifier

Name	Object Identifier
snmpSetSerialNo	::= { snmpSet 1 }

One MIB object is listed in Table 4-42 for the set group.

Table 4-42 Set Group MIB Object

Name	Syntax	Description	Default Value
snmpSetSerialNo	`TestAndIncr`	Specifies an advisory lock used to allow several cooperating SNMPv2 entities, which all act as a manager role, to coordinate their use of the SNMPv2 `Set` operation. This object is used for coarse-grain coordination. To achieve fine-grain coordination, one or more similar objects are appropriately defined within each MIB group. Max-Access: read-write Status: current	none

SNMPv2 Conformance and Compliance Statements

The information on conformance is specific to SNMPv2. Table 4-43 lists the groups.

Table 4-43 SNMPv2 Conformance Groups

Name	Object Identifier
snmpMIBConformance	::= { snmpMIB 2 }
snmpMIBCompliances	::= { snmpMIBConformance 1 }
snmpMIBGroups	::= { snmpMIBConformance 2 }

SNMPv2 Compliance Statements

The compliance statement is used to support SNMPv2. The following are the mandatory groups:

- `snmpGroup`
- `snmpSetGroup`
- `systemGroup`
- `snmpBasicNotificationsGroup`

The compliance object identifier is listed in Table 4-44.

Table 4-44 SNMPv2 Compliance Object Identifier

Name	Object Identifier
s**nmpCommunityGroup**	::= { snmpMIBCompliances 2 }

A list of the MIB objects is listed in Table 4-45.

Table 4-45 SNMPv2 Compliance MIB Objects

Name	Syntax	Description	Default Value
snmpBasicCompliance	none	Specifies the compliance statement for the SNMPv2 entities that implement the SNMPv2 MIB. Status: current	none
snmpCommunityGroup	none	Supports community-based authentication. This group is mandatory for SNMPv2 entities.	none

SNMPv2 Units of Conformance

The object identifiers for each MIB object is listed in Table 4-46.

Table 4-46 SNMPv2 Units of Conformance Object Identifiers

Name	Object Identifier
snmpGroup	::= { snmpMIBGroups 8 }
snmpCommunityGroup	::= { snmpMIBGroups 9 }
snmpSetGroup	::= { snmpMIBGroups 5 }
systemGroup	::= { snmpMIBGroups 6 }
snmpBasicNotificationsGroup	::= { snmpMIBGroups 7 }

The SNMPv2 units of conformance objects are listed in Table 4-47.

Table 4-47 SNMPv2 Objects Used for Units of Conformance

Name	Objects	Description	Default Value
snmpGroup	`snmpInPkts`, `snmpInBadVersions`, `snmpInASNParseErrs`, `snmpSilentDrops`, `snmpProxyDrops`, `snmpEnableAuthenTraps`	Specifies a collection of objects that provides instrumentation and control of an SNMPv2 entity. Status: current	none
snmpCommunityGroup	`snmpInBadCommunity Names`, `snmpInBadCommunity Uses`	Specifies a collection of objects that provides basic instrumentation of a SNMPv2 entity that supports community-based authentication. Status: current	none
snmpSetGroup	`snmpSetSerialNo`	Specifies a collection of objects that allows several cooperating SNMPv2 entities, which all act as a manager role, to coordinate their use of the SNMPv2 `Set` operation. Status: current	none

Table 4-47 SNMPv2 Objects Used for Units of Conformance (continued)

Name	Objects	Description	Default Value
systemGroup	`sysDescr, sysObjectID, sysUpTime, sysContact, sysName, sysLocation, sysServices, sysORLastChange, sysORID, sysORUpTime, sysORDescr`	Defines objects that are common to all managed systems for the system group. Status: current	none
snmpBasicNotifications Group	`coldStart, authenticationFailure`	Specifies the two notifications that an SNMPv2 entity is required to implement. Status: current	none

SNMPv2 Textual Conventions

This section describes the SNMPv2 textual conventions, which reside in the SNMPv2-TC.my file. The names of the textual conventions are specified in the object syntax.

Note Textual conventions do not have any object identifiers.

The SNMPv2 textual conventions are listed in Table 4-48.

Table 4-48 SNMPv2 Textual Conventions MIB Objects

Name	Syntax	Description	Default Value
DisplayString	`OCTET STRING (SIZE (0..255))`	Represents textual information taken from the Network Virtual Terminal (NVT) ASCII character set. For more detailed information, refer to RFC 854. To summarize RFC 854, the following issues are the NVT ASCII: • Use of character codes 0-127 (decimal). • The graphics characters (32-126) are interpreted as US ASCII. • NUL[1], LF[2], CR[3], BEL[4], BS[5], HT[6], VT[7], and FF[8] are defined in RFC 854. • The other 25 codes have no standard interpretation. • The sequence CR LF means new line. • The sequence CR NUL means carriage-return. • The `LF` sequence not preceded by a `CR` means moving to the same column on the next line. • The sequence CR x for any `x` other than LF or NUL is illegal. This also means that a string may end with either CR LF or CR NUL but not with CR). Any object defined using this syntax may not exceed 255 characters in length. Display Hint: `255a` Status: current	none
PhysAddress	`OCTET STRING`	Represents the media or physical-level addresses. Display Hint: `1x` Status current	none
MacAddress	`OCTET STRING (SIZE (6))`	Represents an 802 MAC address represented in the canonical order defined by IEEE 802.1a. For example, if it were transmitted least significant bit first, even though 802.5 (in contrast to other 802.x protocols) requires MAC addresses to be transmitted most significant bit first. Display Hint: `1x` Status: current	none
TruthValue	`INTEGER { true(1), false(2) }`	Represents a boolean value. Status: current	none

Table 4-48 SNMPv2 Textual Conventions MIB Objects (continued)

Name	Syntax	Description	Default Value
TestAndIncr	INTEGER (0..2147483647)	Represents integer-valued information used for atomic operations. When the management protocol is used to specify that an object instance having this syntax is modified, the new value supplied by means of the management protocol must precisely match the value presently held by the instance. If not, the management protocol Set operation fails with an error of inconsistentValue. Otherwise, if the current value is the maximum value of 2^31-1 (2147483647 decimal), the value held by the instance is wrapped to zero; otherwise, the value held by the instance is incremented by one. **Note** Regardless of whether the management protocol Set operation succeeds, the variable- binding in the request and response PDUs are identical. The value of the ACCESS clause for objects having this syntax is either read-write or read-create. When an instance of a columnar object having this syntax is created, any value can be supplied by means of the management protocol. When the network management portion of the system is reinitialized, the value of every object instance having this syntax must either be incremented from the value prior to the reinitialization, or (if the value prior to the reinitialization is unknown) be set to a pseudo-randomly generated value. Status: current	none
AutonomousType	OBJECT IDENTIFIER	Represents an independently extensible type identification value. The value indicates a particular sub-tree contains further MIB definitions, or defines a particular type of protocol or hardware. Status: current	none
VariablePointer	OBJECT IDENTIFIER	Represents a pointer to a specific object instance. For example, the instances can be either sysContact.0 or ifInOctets.3. Status: current	none
RowPointer	OBJECT IDENTIFIER	Represents a pointer to a conceptual row. The value is the name of the instance of the first accessible columnar object in the conceptual row. For example, ifIndex.3 can point to the third row in the ifTable. **Note** If the if ifIndex object is not-accessible, ifDescr.3 can be used instead. Status: current	none

Table 4-48 SNMPv2 Textual Conventions MIB Objects (continued)

Name	Syntax	Description	Default Value
RowStatus	`INTEGER { -- the following two values are states: -- these values may be read or written active(1), notInService(2), -- the following value is a state: -- this value may be read, but not written notReady(3), -- the following three values are -- actions: these values may be written, -- but are never read createAndGo(4), createAndWait(5), destroy(6)}`	Manages the creation and deletion of conceptual rows, and is used as the value of the SYNTAX clause for the status column of a conceptual row (as described in Section 7.7.1 of [2].) The following are the defined values for the status column: • `active`—Indicates that the conceptual row is available for use by the managed device; • `notInService`—Indicates that the conceptual row exists in the agent, but is unavailable for use by the managed device. • `notReady`—Indicates that the conceptual row exists in the agent, but is missing information necessary in order to be available for use by the managed device. • `createAndGo`—Specifies this value is supplied by a management station to create a new instance of a conceptual row. The status is automatically set to active and is made available for use by the managed device. • `createAndWait`—Specifies this value is supplied by a management station to create a new instance of a conceptual row, and is not made available for use by the managed device. • `destroy`—Specifies this value is supplied by a management station to delete all of the instances associated with an existing conceptual row. Five of the six values except `notReady` can be specified in a management protocol `Set` operation. Only three values will be returned in response to a management protocol retrieval operation: `notReady`, `notInService`, or `active`. When queried, an existing conceptual row has the following three states: 1. The row is either available for use by the managed device. The status column contains the value `active`. 2. The row is not available for use by the managed device, though the agent has sufficient information to make it so the status column contains the value `notInService`. 3. The row is not available for use by the managed device, and an attempt to make it so will fail because the agent has insufficient information. The state column contains the value `notReady`. For more detailed information, refer to the SNMPv2-TC.my file. Status: current	none
TimeStamp	`TimeTicks`	Determines the value of the `sysUpTime` object which a specific occurrence happened. The specific occurrence must be defined in the description of any object defined using this type. Status: current	none

Table 4-48 SNMPv2 Textual Conventions MIB Objects (continued)

<table>
<tr><th>Name</th><th>Syntax</th><th>Description</th><th>Default Value</th></tr>
<tr><td>TimeInterval</td><td>INTEGER
(0..2147483647)</td><td>Specifies a period of time that is measured in units of 0.01 sec.
Status: current</td><td>none</td></tr>
<tr><td>DateAndTime</td><td>OCTET STRING
(SIZE (8 | 11))</td><td>Specifies the date and time. For example, the following is displayed for Tuesday May 26, 1992 at 1:30:15 PM EDT:
1992-5-26,13:30:15.0,-4:0
Note If only the local time is known, the time zone information (fields 8 to 10) is not present.
Display Hint: 2d-1d-1d,1d:1d:1d.1d,1a1d:1d
Status: current
The following are the date and time specifications:
<table>
<tr><th>Field</th><th>Octets</th><th>Contents</th><th>Range</th></tr>
<tr><td>1</td><td>1-2</td><td>year</td><td>0..65536</td></tr>
<tr><td>2</td><td>3</td><td>month</td><td>1..12</td></tr>
<tr><td>3</td><td>4</td><td>day</td><td>1..31</td></tr>
<tr><td>4</td><td>5</td><td>hour</td><td>0..23</td></tr>
<tr><td>5</td><td>6</td><td>minutes</td><td>0..59</td></tr>
<tr><td>6</td><td>7</td><td>seconds (use 60 for leap-second)</td><td>0..60</td></tr>
<tr><td>7</td><td>8</td><td>deci-seconds</td><td>0..9</td></tr>
<tr><td>8</td><td>9</td><td>direction from UTC</td><td>'+'/ '-'</td></tr>
<tr><td>9</td><td>10</td><td>hours from UTC</td><td>0..11</td></tr>
<tr><td>10</td><td>11</td><td>minutes from UTC</td><td>0..59</td></tr>
</table></td><td>none</td></tr>
<tr><td>StorageType</td><td>INTEGER {
other(1), -- eh?
volatile(2), --
e.g., in RAM
nonVolatile(3), --
e.g., in NVRAM
permanent(4), --
e.g., partially in
ROM
readOnly(5) --
e.g., completely
in ROM
}</td><td>Describes the memory realization of a conceptual row. A row is volatile(2) and is lost upon reboot. A row is nonVolatile(3), permanent(4), or readOnly(5) and is backed up by stable storage. A row is permanent(4) and is changed but not deleted. A row is readOnly(5) and is not changed nor deleted.
If the value of an object with this syntax is either permanent(4) or readOnly(5), it cannot be modified. Conversely, if the value is either other(1), volatile(2), or nonVolatile(3), it cannot be modified to be permanent(4) or readOnly(5).
Every usage of this textual convention is required to specify the columnar objects that a permanent(4) row at a minimum must allow to be writable.
Status: current</td><td>none</td></tr>
</table>

Table 4-48 SNMPv2 Textual Conventions MIB Objects (continued)

Name	Syntax	Description	Default Value
TDomain	OBJECT IDENTIFIER	Denotes a kind of transport service. Some possible values, such as snmpUDPDomain, are defined in the Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2). Status: current	none
TAddress	OCTET STRING (SIZE (1..255))	Denotes a transport service address. For snmpUDPDomain, a TAddress is six octets long, the initial 4 octets containing the IP-address in network-byte order, and the last 2 containing the UDP port in network-byte order. For more detailed information on snmpUDPDomain, refer to Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2). Status: current	none

1. NUL = NULL
2. LF = line feed
3. CR = carriage return
4. BEL = BELL
5. BS = back space
6. HT = horizontal tab
7. VT = Virtual terminal
8. FF = form feed

Cisco VSI Controller MIB Objects

This section describes the individual MIB objects used to configure an ATM switch to have Virtual Switch Interface (VSI) Controller information. The VSI Controller resides in the CISCO-VSI-CONTROLLER-MIB.my file.

The Cisco VSI controller MIB objects include:

- Cisco VSI Controller Terminology
- Cisco VSI Controller Textual Conventions
- Cisco VSI Controller Configuration Table
- Cisco VSI Controller Conformance and Compliance Information

Cisco VSI Controller Terminology

The terms used for the VSI Controller are listed in Table 4-49.

Table 4-49 Terms Used for the VSI Controller

Term	Description
VSI	Describes the hardware-independent switch control protocol. This allows a switch, for example, a node, to be controlled by a multiple controllers such as PNNI and Label Switch Controller (LSC). These control planes are internal or external to the switch. The VSI interface defines the messages, and associated functions that allow communication between the controller and the switch. This interface is expected to support all types of connections, for example, voice, data, Frame Relay, and ATM, used for Permanent Virtual Circuit (PVC), Soft Permanent Virtual Connection (SPVC), and Switched Virtual Circuit (SVC).
VSI Master	Requests connections and receives switch-generic information for the software component, and controls one or more VSI Slaves. This can run on the switch or a dedicated controller platform, which is the master module. It performs the interface to the higher layer networking software and handles all VSI-related functions.
VSI Slave	Converts generic connection requests into hardware-specific requests, and hardware-specific information into generic information for the software component. VSI Slave runs on the switch. A centralized slave has a single point of control for making connections and controlling interfaces, while a distributed slave allows for multiple slaves to coexist on the same switch.
Controller	Describes the software and hardware that manages topology and network resources. It also performs the VSI Master function. The Controller performs source routing for end-to-end SVCs, which includes general call acceptance Generic Connection Admission Control (GCAC) and setup calls with other controllers. PNNI and Multiprotocol Label Switching (MPLS) are examples for the Controller.
Controller Shelf	Specifies a Controller Shelf that is a switch containing at least one VSI Controller, which is controlling a different switch. Also, it will contain local controllers.

Cisco VSI Controller Textual Conventions

The names of the textual conventions are specified in the object syntax.

A list of the MIB objects is listed in Table 4-50.

Note Textual conventions do not contain object identifiers.

Table 4-50 Cisco VSI Controller Textual Convention MIB Objects

Name	Description	Syntax	Default Value
CvcControllerShelf Location	Specifies the location of the controller shelf. The values include: • `internal(1)`—Specifies that the controller resides on the same shelf as the switch. • `external(2)`—Specifies that the controller resides on the external platform. The controller shelf is connected to the switch by an ATM link. Status: current	`INTEGER{internal(1), external(2)}`	none
CvcControllerType	Describes the type of controller that is a VSI Master. The values include: • `par(1)`—Portable Auto Route (PAR). This is a VSI Master controller that implements Cisco proprietary protocol for network routing and topology in a network, which contains only Cisco switches. • `pnni(2)`—Private Network-to-Network Interface (PNNI) controller. The PNNI protocol is used between private ATM Switches and between groups of ATM switches. This protocol is defined for distributing topology information between switches and clusters of switches. • `lsc(3)`—Label Switch Controller (LSC). The LSC implements the MPLS protocol. The LSC is a router which is capable of controlling the operation of a separate ATM switch, so that both of them function as a single ATM-LSR (ATM Label Switch Router). LSC controls the operation of the ATM switch using a Switch Control Protocol, which allows the LSC to setup and remove cross-connections on the ATM switch. It is used to discover the configuration and capabilities of the controlled switch, and to gather statistics from the controlled switch. Status: current	`INTEGER {par(1), pnni(2), lsc(3)}`	none

Cisco VSI Controller Configuration Table

The object identifiers are listed in Table 4-51.

Table 4-51 VSI Controller Configuration Table Object Identifiers

Name	Object Identifier
cvcConfTable	::= { cvcConfController 1 }
cvcConfEntry	::= { cvcConfTable 1 }
cvcConfControllerID	::= { cvcConfEntry 1}
cvcConfControllerType	::= { cvcConfEntry 2}
cvcConfControllerShelfLocation	::= { cvcConfEntry 3}
cvcConfControllerLocation	::= { cvcConfEntry 4}
cvcConfControllerName	::= { cvcConfEntry 5 }
cvcConfVpi	::= { cvcConfEntry 6 }
cvcConfVci	::= { cvcConfEntry 7 }
cvcConfRowStatus	::= { cvcConfEntry 8 }

A list of the MIB objects is listed in Table 4-52.

Table 4-52 Cisco VSI Controller Configuration Table MIB Objects

Name	Syntax	Description	Default Value
cvcConfTable	`SEQUENCE OF CvcConfEntry`	Contains the entries for the VSI controllers. This table is used to inform the VSI slaves about the existence of the VSI controllers, and how the VSI slaves can reach the controller. When an entry is created and activated, the information in these entries is advertised to all the VSI slaves, using a system-dependent implementation. Max-Access: not-accessible Status: current	none
cvcConfEntry	`CvcConfEntry`	Provides an entry for a VSI Controller. The entries in this table are created by setting the `cvcConfRowStatus` object to `createAndGo(4)`. The entries in this table are deleted by setting the `cvcConfRowStatus` object to `destroy(6)`. The entries are created, modified, and deleted through the CLI. Max-Access: not-accessible Status: current The index is `cvcConfControllerID`.	none

Table 4-52 Cisco VSI Controller Configuration Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cvcConfControllerID	`Integer32 (1..2147483647)`	Specifies the unique value for the VSI Controller, for example, VSI Master. The VSI Slave uses this value in the message to identify the VSI Master Controller. Max-Access: not-accessible Status: current	none
cvcConfController Type	`CvcControllerType`	Identifies the controller type. This object cannot be modified if the associated `cvcConfRowStatus` is equal to `active(1)`. Max-Access: read-create Status: current	none
cvcConfController ShelfLocation	`CvcControllerShelfLocation`	Identifies the location of the controller shelf. This object is set only during row creation. Max-Access: read-create Status: current	none
cvcConfController Location	`Integer32 (1..2147483647)`	Identifies the location of the controller. The `cvcConfControllerLocation` object can contain the logical slot number of the module where the controller is running on the same shelf as the switch. This object can contain the value of the interface on the module where the controller is running on an external shelf connected to the switch. If the associated `cvcConfRowStatus` is equal to `active(1)`, this object cannot be modified. Max-Access: read-create Status: current	none
cvcConfController Name	`DisplayString`	Specifies the name chosen by the user for the VSI Controller. If the user does not set the value for this object, it contains an octet string of length zero. `cvcConfControllerName` object cannot be modified if the associated `cvcConfRowStatus` is equal to `active(1)`. Max-Access: read-create Status: current	none
cvcConfVpi	`Integer32 (0..4095)`	Specifies the VPI that is used for connecting to the controller, which is external to the switch. This object has significance only if `cvcConfControllerShelfLocation` is `external(2)`. The `cvcConfVpi` object cannot be modified if the associated `cvcConfRowStatus` is equal to `active(1)`. Max-Access: read-create Status: current	none

Table 4-52 Cisco VSI Controller Configuration Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cvcConfVci	`Integer32 (32..65535)`	Specifies the start value of VCI that is used for connecting to the controller which is external to the switch. This object has significance only if `cvcConfControllerShelfLocation` is `external(2)`. `cvcConfVci` object cannot be modified if the associated `cvcConfRowStatus` is equal to `active(1)`. Max-Access: read-create Status: current	none
cvcConfRowStatus	`RowStatus`	Adds, deletes, and modifies the controller configuration. The row is created by setting this object to `createAndGo(4)`. The row is deleted by setting this object to `destroy(6)`. When this object contains value `active(1)`, the objects in the row are not modified. Max-Access: read-create Status: current	none

Cisco VSI Controller Conformance and Compliance Information

The object identifiers are listed in Table 4-53 for conformance.

Table 4-53 Cisco VSI Controller Conformance Groups

Name	Object Identifier
cvcMIBConformance	::= {ciscoVSIControllerMIB 3}
cvcMIBCompliances	::= {cvcMIBConformance 1}
cvcMIBGroups	::= {cvcMIBConformance 2}

Cisco VSI Controller Compliance Statement

The compliance object identifier is listed in Table 4-54.

Table 4-54 Cisco VSI Controller Object Identifier

Name	Object Identifier
cvcConfGroupExternal	::= {cvcMIBCompliances 1}

The objects for compliance are listed in Table 4-55.

Table 4-55 Cisco VSI Controller Compliance Objects

Name	Groups	Description	Default Value
cvcMIBCompliance	cvcConfGroup	Describes the compliance statement for the Cisco VSI Controller group. Status: current	none
cvcConfGroupExternal	none	Specifies this group is required only for controllers running on a shelf external to the switch.	none

Cisco VSI Controller Units of Conformance

The object identifiers for Cisco VSI Controller units of conformance are listed in Table 4-56.

Table 4-56 Cisco VSI Controller Units of Conformance Object Identifiers

Name	Object Identifier
cvcConfGroup	::= { cvcMIBGroups 1}
cvcConfGroupExternal	::= { cvcMIBGroups 2}

The Cisco VSI Controller units of conformance objects are listed in Table 4-57.

Table 4-57 Cisco VSI Controller Objects Used for Units of Conformance

Name	Groups	Description	Default Value
cvcConfGroup	cvcConfControllerType, cvcConfControllerShelfLocation, cvcConfControllerLocation, cvcConfControllerName, cvcConfRowStatus	Configures the VSI controllers that run on the same shelf as the switch. Status: current	none
cvcConfGroupExternal	cvcConfVpi, cvcConfVci	Configures the VSI controllers that run on the shelf external to the switch. Status: current	none

CHAPTER 5

PNNI MIB Objects

This chapter describes the MIB objects used for PNNI.

Contents of this chapter include:

- ATM MIB Objects
- ATM PNNI MIB Objects
- Cisco WAN SVC MIB Objects

Note PNNI MIB objects are supported by PXM45.

ATM MIB Objects

This section describes the individual MIB objects that comprise the ATM MIB file. The ATM MIB resides in the ATM-MIB.my file.

The ATM MIB objects include:

- ATM Interface Configuration Parameters Table
- ATM Interface DS3 Physical Layer Convergence Procedure Table
- ATM Interface Transmission Convergence Sublayer Table

ATM Interface Configuration Parameters Table

This table contains ATM specific configuration information that is associated with an ATM interface beyond those supported using the `ifTable`.

The object identifier for each MIB object is listed in Table 5-1.

Table 5-1 ATM Interface Configuration Parameters Table Object Identifiers

Name	Object Identifier
atmInterfaceConfTable	::= { atmMIBObjects 2 }
atmInterfaceConfEntry	::= { atmInterfaceConfTable1}
atmInterfaceMaxVpcs	::= { atmInterfaceConfEntry 1}
atmInterfaceMaxVccs	::= { atmInterfaceConfEntry 2}

Table 5-1 ATM Interface Configuration Parameters Table Object Identifiers (continued)

Name	Object Identifier
atmInterfaceConfVpcs	::= { atmInterfaceConfEntry 3}
atmInterfaceConfVccs	::= { atmInterfaceConfEntry 4}
atmInterfaceMaxActiveVpiBits	::= { atmInterfaceConfEntry 5}
atmInterfaceMaxActiveVciBits	::= { atmInterfaceConfEntry 6}
atmInterfaceIlmiVpi	::= { atmInterfaceConfEntry 7}
atmInterfaceIlmiVci	::= { atmInterfaceConfEntry 8}
atmInterfaceAddressType	::= { atmInterfaceConfEntry 9 }
atmInterfaceAdminAddress	::= { atmInterfaceConfEntry 10 }
atmInterfaceMyNeighborIpAddress	::= { atmInterfaceConfEntry 11 }
atmInterfaceMyNeighborIfName	::= { atmInterfaceConfEntry 12 }
atmInterfaceCurrentMaxVpiBits	::= { atmInterfaceConfEntry 13 }
atmInterfaceCurrentMaxVciBits	::= { atmInterfaceConfEntry 14 }
atmInterfaceSubscrAddress	::= { atmInterfaceConfEntry 15 }

The MIB objects are listed in Table 5-2.

Table 5-2 ATM Interface Configuration Parameters Table MIB Objects

Name	Syntax	Description	Default Value
atmInterfaceConfTable	`Sequence of AtmInterfaceConfEntry`	Contains the ATM local interface configuration parameters. This table is used for one entry per ATM interface port. Max-Access: not-accessible Status: current	none
atmInterfaceConfEntry	`AtmInterfaceConfEntry`	Contains the ATM interface configuration parameters and state variables. This list is indexed by `ifIndex` values of ATM interfaces. Max-Access: not-accessible Status: current The `ifindex` is listed as the index.	none
atmInterfaceMaxVpcs	`INTEGER (0..4096)`	Specifies the maximum number of Virtual Path Links (VPC), for example, Permanent Virtual Path Circuits (PVPC) and Switched Virtual Path Circuits (SVPC), which are supported at this ATM interface. At the ATM UNI, the maximum number of VPCs used for PVPCs and SVPCs range from 0 to 256. Max-Access: read-write Status: current	none

Table 5-2 ATM Interface Configuration Parameters Table MIB Objects (continued)

Name	Syntax	Description	Default Value
atmInterfaceMaxVccs	INTEGER (0..65536)	Specifies the maximum number of Virtual Channel Connections (VCC). PVCC and SVCC are supported at the ATM interface. Max-Access: read-write Status: current	none
atmInterfaceConfVpcs	INTEGER (0..4096)	Specifies the number of VPC. PVPC, Soft PVPC, and SVPC are established at the ATM interface. At the ATM UNI, the configured number of VPCs used for PVPCs and SVPCs range from 0 to 256. Max-Access: read-only Status: current	none
atmInterfaceConfVccs	INTEGER (0..65536)	Specifies the number of VCC. PVCC, Soft PVCC, and SVCC are established at the ATM interface. Max-Access: read-only Status: current	none
atmInterfaceMaxActive VpiBits	INTEGER (0..12)	Specifies the maximum number of active Virtual Path Identifier (VPI) bits configured for this ATM interface. At the ATM UNI, the maximum number of active VPI bits configured ranges from 0 to 8. Max-Access: read-write Status: current	none
atmInterfaceMaxActive VciBits	INTEGER (0..16)	Specifies the maximum number of active Virtual Channel Identifier (VCI) bits configured for this ATM interface. Max-Access: read-write Status: current	none
atmInterfaceIlmiVpi	AtmVpIdentifier	Specifies the VPI value of the VCC that supports the Interim Local Management Interface (ILMI) at this ATM interface. If the values of atmInterfaceIlmiVpi and atmInterfaceIlmiVci are both equal to 0, ILMI is not supported at this ATM interface. Max-Access: read-write Status: current	0
atmInterfaceIlmiVci	AtmVcIdentifier	Specifies the VCI value of the VCC that supports the ILMI at this ATM interface. If the values of atmInterfaceIlmiVpi and atmInterfaceIlmiVci are both equal to 0, ILMI is not supported at this ATM interface. Max-Access: read-write Status: current	16

Table 5-2 ATM Interface Configuration Parameters Table MIB Objects (continued)

Name	Syntax	Description	Default Value
atmInterfaceAddress Type	`INTEGER { private (1), nsapE164 (2), nativeE164 (3),other (4)}`	Specifies the type of primary ATM address that is configured for this ATM interface. Max-Access: read-only Status: deprecated	none
atmInterfaceAdmin Address	`AtmAddr`	Specifies the primary address assigned for administrative purposes, for example, an address associated with the service provider side of a public network UNI. Thus, the value of this address corresponds with the value of `ifPhysAddress` at the host side. If this interface has no assigned administrative address, or the address used for administrative purposes is the same as that used for `ifPhysAddress`, this is an octet string of zero length. Max-Access: read-only Status: deprecated **Note** The `atmInterfaceAdminAddress` object is replaced by `atmInterfaceSubscrAddress`.	none
atmInterfaceMyNeighbor IpAddress	`IpAddress`	Specifies the IP address of the neighbor system that is connected to the far end of this interface. An NMS can send SNMP messages as IP datagrams. These messages are sent to UDP port 161 to access network management information, which concerns the operation of the system. **Note** The value of this object is obtained in different ways, for example, by manual configuration, or through ILMI interaction with the neighbor system. Max-Access: read-only Status: current	none

Table 5-2 ATM Interface Configuration Parameters Table MIB Objects (continued)

Name	Syntax	Description	Default Value
atmInterfaceMyNeighborIfName	`DisplayString`	Specifies the textual name of the interface on the neighbor system on the far end of this interface and to where this interface connects. If the neighbor system is manageable through SNMP and supports the object `ifName`, the value of this object must be identical with `ifName` for the `ifEntry` of the lowest-level physical interface for this port. If this interface does not have a textual name, the value of this object is a zero length string. **Note** The value of this object can be obtained in different ways, for example, by manual configuration, or through ILMI interaction with the neighbor system. Max-Access: read-only Status: current	none
atmInterfaceCurrentMaxVpiBits	`INTEGER (0..12)`	Determines the maximum number of VPI Bits that are currently used at this ATM interface. The value is the minimum of `atmInterfaceMaxActiveVpiBits`, and the `atmInterfaceMaxActiveVpiBits` of this interface UNI or NNI peer. If the interface does not negotiate with the peer to determine the number of VPI Bits that are used on the interface, the value of this object must equal `atmInterfaceMaxActiveVpiBits`. Max-Access: read-only Status: current	none

Table 5-2 ATM Interface Configuration Parameters Table MIB Objects (continued)

Name	Syntax	Description	Default Value
atmInterfaceCurrentMaxVciBits	`INTEGER (0..16)`	Determines the maximum number of VCI Bits that are currently used at this ATM interface. The value is the minimum of `atmInterfaceMaxActiveVciBits`, and the `atmInterfaceMaxActiveVciBits` of this interface UNI or NNI peer. If the interface does not negotiate with the peer to determine the number of VCI Bits that is used on the interface, the value of this object must equal `atmInterfaceMaxActiveVciBits`. Max-Access: read-only Status: current	none
atmInterfaceSubscrAddress	`AtmAddr`	Specifies the identifier that is assigned by a service provider to the network side of a public network UNI. If this interface has no assigned service provider address for other interfaces, this is an octet string of zero length. Max-Access: read-write Status: current	none

ATM Interface DS3 Physical Layer Convergence Procedure Table

This table contains the DS3 Physical Layer Convergence Procedure (PLCP) configuration and state parameters for the ATM interfaces, which use DS3 PLCP for carrying ATM cells over DS3.

The object identifier for each MIB object is listed in Table 5-3.

Table 5-3 ATM Interface DS3 PLCP Table Object Identifiers

Name	Object Identifier
atmInterfaceDs3PlcpTable	::= { atmMIBObjects 3}
atmInterfaceDs3PlcpEntry	::= { atmInterfaceDs3PlcpTable 1}
atmInterfaceDs3PlcpSEFSs	::= { atmInterfaceDs3PlcpEntry 1}
atmInterfaceDs3PlcpAlarmState	::= { atmInterfaceDs3PlcpEntry 2}
atmInterfaceDs3PlcpUASs	::= { atmInterfaceDs3PlcpEntry 3}

The MIB objects are listed in Table 5-4.

Table 5-4 ATM Interface DS3 PLCP Table MIB Objects

Name	Syntax	Description	Default Value
atmInterfaceDs3Plcp Table	`Sequence of AtmInterfaceDs3PlcpEntry`	Contains the ATM interface DS3 PLCP parameters and state variables that are used for one entry per ATM interface port. Max-Access: not-accessible Status: current	none
atmInterfaceDs3Plcp Entry	`AtmInterfaceDs3PlcpEntry`	Contains the DS3 PLCP parameters and state variables at the ATM interface. It is indexed by the `ifIndex` value of the ATM interface. Max-Access: not-accessible Status: current The index contains `ifIndex`.	none
atmInterfaceDs3Plcp SEFSs	`Counter32`	Specifies the number of DS3 PLCP Severely Errored Framing Seconds (SEFS). Each SEFS represents a one-second interval that contains one or more severely Errored Framing (SEF) events. Max-Access: read-only Status: current	none
atmInterfaceDs3Plcp AlarmState	`Integer {noAlarm (1), receivedFarEndAlarm (2), incomingLOF(3)}`	Indicates if there is an alarm present for the DS3 PLCP. The following are the values: • `receivedFarEndAlarm`—Specifies that the DS3 PLCP has received an incoming Yellow Signal. • `incomingLOF`—Specifies that the DS3 PLCP has declared a loss of frame (LOF) failure condition. • `noAlarm`—Specifies that no alarms are present. A transition from the failure to the no alarm state occurs when no defects (for example, LOF) are received for more than 10 seconds. Max-Access: read-only Status: current	none
atmInterfaceDs3Plcp UASs	`Counter32`	Specifies the counter that is associated with the number of unavailable seconds encountered by PLCP. Max-Access: read-only Status: current	none

ATM Interface Transmission Convergence Sublayer Table

This table contains the transmission convergence (TC) sublayer configuration and state parameters of the ATM interfaces, which use TC sublayer to carry ATM cells over SONET/SDH or DS3.

The object identifier for each MIB object is listed in Table 5-5.

Table 5-5 ATM Interface TC Sublayer Table Object Identifiers

Name	Object Identifier
atmInterfaceTCTable	::= { atmMIBObjects 4}
atmInterfaceTCEntry	::= { atmInterfaceTCTable 1}
atmInterfaceOCDEvents	::= { atmInterfaceTCEntry 1}
atmInterfaceTCAlarmState	::= { atmInterfaceTCEntry 2}

The MIB objects are listed in Table 5-6.

Table 5-6 ATM Interface TC Sublayer Table MIB Objects

Name	Syntax	Description	Default Value
atmInterfaceTCTable	`Sequence of AtmInterfaceTCEntry`	Contains the ATM interface TC sublayer parameters and state variables that are used for one entry per ATM interface port. Max-Access: not-accessible Status: current	none
atmInterfaceTCEntry	`AtmInterfaceTCEntry`	Contains the TC subpolar parameters and state variables at the ATM interface. It is indexed by the `ifIndex` value of the ATM interface. Max-Access: not-accessible Status: current The index contains `ifIndex`.	none

Table 5-6 ATM Interface TC Sublayer Table MIB Objects (continued)

Name	Syntax	Description	Default Value
atmInterfaceOCD Events	`Counter32`	Determines the number of times the Out of Cell Delineation (OCD) events occur. If seven consecutive ATM cells have Header Error Control (HEC) violations, an OCD event occurs. A high number of OCD events can indicate a problem with the TC sublayer. Max-Access: read-only Status: current	none
atmInterfaceTCAlarm State	`Integer{noAlarm (1), lcdFailure(2)}`	Indicates if there is an alarm present for the TC sublayer. The following are the values used for the `atmInterfaceTCAlarmState`: • `lcdFailure(2)`—Indicates that the TC sublayer is currently in the Loss of Cell Delineation (LCD) defect maintenance state. • `noAlarm(1)`—Indicates that the TC sublayer is currently not in the LCD defect maintenance state. Max-Access: read-only Status: current	none

ATM PNNI MIB Objects

This section describes the individual MIB objects that comprise ATM PNNI MIB file. The ATM PNNI MIB is defined in the PNNI-MIB.my file.

The ATM PNNI MIB objects include:

- Base Group
- Private Network-to-Network Interface Node Table
- Peer Group Leader Election Table
- Initial Timer Values Table
- Nodal SVCC-Based RCC Variables Table
- Scope Mapping Table
- Summary Address Table
- Link Table
- ATM PNNI MIB Conformance and Compliance Statements

Base Group

The object identifier for each MIB object is listed in Table 5-7.

Table 5-7 Base Group Object Identifiers

Name	Object Identifier
pnniHighestVersion	::= { pnniBaseGroup 1 }
pnniLowestVersion	::= { pnniBaseGroup 2 }
pnniDtlCountOriginator	::= { pnniBaseGroup 3 }
pnniDtlCountBorder	::= { pnniBaseGroup 4 }
pnniCrankbackcountOriginator	::= { pnniBaseGroup 5 }
pnniCrankbackCountBorder	::= { pnniBaseGroup 6 }
pnniAltRouteCountOriginator	::= { pnniBaseGroup 7 }
pnniAltRouteCountBorder	::= { pnniBaseGroup 8 }
pnniRouteFailCountOriginator	::= { pnniBaseGroup 9 }
pnniRouteFailCountBorder	::= { pnniBaseGroup 10 }
pnniRouteFailUnreachableOriginator	::= { pnniBaseGroup 11 }
pnniRouteFailUnreachableBorder	::= { pnniBaseGroup 12 }

The MIB objects are listed in Table 5-8.

Table 5-8 Base Group MIB Objects

Name	Syntax	Description	Default Value
pnniHighestVersion	`PnniVersion`	Specifies the highest version of the PNNI protocol that the software is capable of executing. Refer to *ATM Forum PNNI 1.0*, Section 5.6.1. Max-Access: read-only Status: current	none
pnniLowestVersion	`PnniVersion`	Specifies the lowest version of the PNNI Protocol that the software is capable of executing. Refer to *ATM Forum PNNI 1.0*, Section 5.6.1. Max-Access: read-only Status: current	none

Table 5-8 Base Group MIB Objects (continued)

Name	Syntax	Description	Default Value
pnniDtlCountOriginator	Counter32	Determines the total number of Designed Transit List (DTL) stacks that originated as the DTLOriginator. The stacks are placed into signaling messages. This includes the initial DTL stacks computed by this system as well as any alternate route (second, third choice, and so forth). DTL stacks are computed to respond to crankbacks. Max-Access: read-only Status: current	none
pnniDtlCountBorder	Counter32	Determines the number of partial DTL stacks added into signaling messages as an entry border node. This includes the partial DTL stacks computed by this system as well as any alternate route (second, third choice, and so forth). The partial DTL stacks are computed to respond to crankbacks. Max-Access: read-only Status: current	none
pnniCrankbackCountOriginator	Counter32	Determines the total number of connection setup messages. It includes DTL stacks that cranked back to this switching system at all levels of the hierarchy. Max-Access: read-only Status: current	none
pnniCrankbackCountBorder	Counter32	Determines the total number of connection setup messages. It includes the DTLs that are added by this switching system as an entry border node, and have cranked back to this switching system at all levels of the hierarchy. This count does not include crankbacks where the switching system was not at the crankback destination. Only the crankbacks that are directed to this switching system are counted here. Max-Access: read-only Status: current	none
pnniAltRouteCountOriginator	Counter32	Determines the total number of alternate DTL stacks. These stacks are computed and placed into signaling messages as the DTLOriginator. Max-Access: read-only Status: current	none
pnniAltRouteCountBorder	Counter32	Determines the total number of alternate partial DTL stacks that is computed and placed into signaling messages as an entry border node. Max-Access: read-only Status: current	none

Table 5-8 Base Group MIB Objects (continued)

Name	Syntax	Description	Default Value
pnniRouteFailCountOriginator	Counter32	Determines the total number of times the switching system failed to compute a viable DTL stack as the DTLOriginator for a call. It indicates the number of times a call was cleared from the switching system due to originator routing failure. Max-Access: read-only Status: current	none
pnniRouteFailCountBorder	Counter32	Determines the total number of times the switching system failed to compute a viable partial DTL stack as an entry border node for a call. It indicates the number of times a call was either cleared or cranked back from the switching system due to a border routing failure. Max-Access: read-only Status: current	none
pnniRouteFailUnreachableOriginator	Counter32	Determines the total number of times the switching system failed to compute a viable DTL stack as the DTLOriginator. Because the destination was unreachable, calls are cleared with cause #2 specified transit network unreachable or cause #3 destination unreachable in the cause IE. Max-Access: read-only Status: current	none
pnniRouteFailUnreachableBorder	Counter32	Determines the total number of times the switching system failed to compute a viable partial DTL stack as an entry border node. Because the target of the path calculation was unreachable, calls are cleared or cranked back with cause #2 specified transit network unreachable or cause #3 destination unreachable in the cause IE. Max-Access: read-only Status: current	none

Private Network-to-Network Interface Node Table

The MIB object identifier used for the Private Network-to-Network Interface (PNNI) node table is listed in Table 5-9.

Table 5-9 PNNI Node Table Object Identifiers

Name	Object Identifier
pnniNodeTable	::= { pnniMIBObjects 2 }
pnniNodeEntry	::= { pnniNodeTable 1 }
pnniNodeIndex	::= { pnniNodeEntry 1 }

Table 5-9 PNNI Node Table Object Identifiers (continued)

Name	Object Identifier
pnniNodeLevel	::= { pnniNodeEntry 2 }
pnniNodeId	::= { pnniNodeEntry 3 }
pnniNodeLowest	::= { pnniNodeEntry 4 }
pnniNodeAdminStatus	::= { pnniNodeEntry 5 }
pnniNodeOperStatus	::= { pnniNodeEntry 6 }
pnniNodeDomainName	::= { pnniNodeEntry 7 }
pnniNodeAtmAddress	::= { pnniNodeEntry 8 }
pnniNodePeerGroupId	::= { pnniNodeEntry 9 }
pnniNodeRestrictedTransit	::= { pnniNodeEntry 10 }
pnniNodeComplexRep	::= { pnniNodeEntry 11 }
pnniNodeRestrictedBranching	::= { pnniNodeEntry 12}
pnniNodeDatabaseOverload	::= { pnniNodeEntry 13 }
pnniNodePtses	::= { pnniNodeEntry 14 }
pnniNodeRowStatus	::= { pnniNodeEntry 15 }

The MIB objects are listed in Table 5-10.

Table 5-10 PNNI Node Table MIB Objects

Name	Syntax	Description	Default Value
pnniNodeTable	`Sequence of PnniNodeEntry`	Collects the attributes that affect the operation of a PNNI logical node. A single row in this table is needed for each PNNI peer group, which the managed system is expected or eligible to become a member of. Refer to *ATM Forum PNNI 1.0*, Annex F. Max-Access: not-accessible Status: current	none
pnniNodeEntry	`PnniNodeEntry`	Contains information about a PNNI logical node. Refer to *ATM Forum PNNI 1.0*, Annex F. Max-Access: not-accessible Status: current	none
pnniNodeIndex	`PnniNodeIndex`	Assigns a value to a node that uniquely identifies it in the MIB. Max-Access: not-accessible Status: current	none

Table 5-10 PNNI Node Table MIB Objects (continued)

Name	Syntax	Description	Default Value
pnniNodeLevel	PnniLevel	Specifies the level of PNNI hierarchy where this node exists. This attribute determines the default node ID and the default peer group ID. This object can be written only when pnniNodeAdminStatus has the value down. Refer to *ATM Forum PNNI 1.0*, Section 5.3.1, Annex F. Max-Access: read-create Status: current	96
pnniNodeId	PnniNodeId	Specifies the value this node uses to represent itself. This object can be written only when pnniNodeAdminStatus has the value down. If pnniNodeLowest is true, the default node ID takes the form defined in the *ATM Forum PNNI 1.0*, Section 5.3.3 for lowest level nodes, with the first octet equal to pnniNodeLevel, the second octet equal to 160, and the last 20 octets equal to pnniNodeAtmAddress. If pnniNodeLowest is false, the default node ID takes the form defined in the *ATM Forum PNNI 1.0*, Section 5.3.3. If the logical group nodes, with the first octet equal to pnniNodeLevel, the next 14 octets equal to the value of pnniNodePeerGroupId for the child node whose election as PGL causes this LGN to be instantiate. The next 6 octets equal to the ESI of pnniNodeAtmAddress, and the last octet equal to 0. Refer to *ATM Forum PNNI 1.0*, Section 5.3.3, Annex F. Max-Access: read-create Status: current	none
pnniNodeLowest	TruthValue	Indicates whether this node acts on the low-level node or whether this node is a logical group node that becomes active when one of the other nodes become a peer group leader. The value false must not be used with nodes that are not PGL/LGN capable. This object can be written only when pnniNodeAdminStatus has the value down. Max-Access: read-create Status: current	true
pnniNodeAdminStatus	INTEGER { up(1), down(2) }	Indicates whether the administrative status of the node is up (the node is allowed to become active) or down (the node is forced to be inactive). When pnniNodeAdminStatus is down, pnniNodeOperStatus must also be down. Max-Access: read-create Status: current	up

Table 5-10 PNNI Node Table MIB Objects (continued)

Name	Syntax	Description	Default Value
pnniNodeOperStatus	`INTEGER { up(1), down(2) }`	Indicates whether the node is active or operational. When the value is `down`, all state is cleared from the node and the node is not communicating with any of its neighbor nodes. Max-Access: read-only Status: current	none
pnniNodeDomainName	`DisplayString`	Specifies the name of the PNNI routing domain where the node participates. All low-level PNNI nodes with the same `pnniNodeDomainName` are presumed to be connected. Max-Access: read-create Status: current	none
pnniNodeAtmAddress	`PnniAtmAddr`	Determines the ATM End System Address for this node. Remote systems that need to exchange PNNI protocol packets with the node should direct packets or calls to this address. This attribute is written only when `pnniNodeAdminStatus` has the value `down`. Refer to *ATM Forum PNNI 1.0*, Section 5.2.2. Max-Access: read-create Status: current	none
pnniNodePeerGroupId	`PnniPeerGroupId`	Specifies the Peer Group Identifier of the peer group that the given node needs to become a member. The default value of this attribute has the first octet equal to `pnniNodeLevel`. The next `pnniNodeLevel` bits equal to the `pnniNodeLevel` bits starting from the third octet of `pnniNodeId`. The remainder are padded with zeros. This object is written only when `pnniNodeAdminStatus` has the value down. Refer to *ATM Forum PNNI 1.0*, Section 5.3.2, Annex F. Max-Access: read-create Status: current	none
pnniNodeRestrictedTransit	`TruthValue`	Specifies whether the node is restricted to not allow support of SVC transiting this node. This attribute determines the setting of the restricted transit bit in the nodal information group. Refer to *ATM Forum PNNI 1.0*, Section 5.8.1.2.3. Max-Access: read-create Status: current	`false`

Table 5-10 PNNI Node Table MIB Objects (continued)

Name	Syntax	Description	Default Value
pnniNodeComplexRep	`TruthValue`	Specifies whether the node uses the complex node representation. A value of `true` indicates that the complex node representation is used; whereas, a value of `false` indicates that the simple node representation is used. This attribute determines the setting of the nodal representation bit in the nodal information group. Refer to *ATM Forum PNNI 1.0*, Section 5.8.1.2.3. Max-Access: read-create Status: current	none
pnniNodeRestrictedBranching	`TruthValue`	Indicates whether the node is able to support additional point-to-multipoint branches. A value of `false` indicates that additional branches are supported; a value of `true` indicates that additional branches are not supported. This attribute reflects the setting of the restricted branching bit in the nodal information group. Refer to *ATM Forum PNNI 1.0*, Section 5.8.1.2.3. Max-Access: read-only Status: current	none
pnniNodeDatabaseOverload	`TruthValue`	Specifies whether the node is currently operating in topology database overload state. This attribute has the same value as the non-transit for PGL Election bit in the nodal information group originated by this node. Refer to *ATM Forum PNNI 1.0*, Section 5.8.1.2.3. Max-Access: read-only Status: current	none
pnniNodePtses	`Gauge32`	Gauges the total number of PNNI Topology State Elements (PTSE) currently in the node topology database(s). Max-Access: read-only Status: current	none
pnniNodeRowStatus	`RowStatus`	Creates, deletes, activates, and deactivates a node. Max-Access: read-create Status: current	none

Peer Group Leader Election Table

The object identifier for each MIB object used for the Peer Group Leader (PGL) election table is listed in Table 5-11.

Table 5-11 PGL Election Table Object Identifiers

Name	Object Identifier
pnniNodePglTable	::= { pnniMIBObjects 3 }
pnniNodePglEntry	::= { pnniNodePglTable 1 }
pnniNodePglLeadershipPriority	::= { pnniNodePglEntry 1 }
pnniNodeCfgParentNodeIndex	::= { pnniNodePglEntry 2 }
pnniNodePglInitTime	::= { pnniNodePglEntry 3 }
pnniNodePglOverrideDelay	::= { pnniNodePglEntry 4 }
pnniNodePglReelectTime	::= { pnniNodePglEntry 5 }
pnniNodePglState	::= { pnniNodePglEntry 6 }
pnniNodePreferredPgl	::= { pnniNodePglEntry 7 }
pnniNodePeerGroupLeader	::= { pnniNodePglEntry 8 }
pnniNodePglTimeStamp	::= { pnniNodePglEntry 9 }
pnniNodeActiveParentNodeId	::= { pnniNodePglEntry 10 }

The MIB objects are listed in Table 5-12.

Table 5-12 PGL Election Table MIB Objects

Name	Syntax	Description	Default Value
pnniNodePglTable	`Sequence of PnniNodePglEntry`	Specifies the peer group leader election information for a PNNI node. Refer to *ATM Forum PNNI 1.0*, Section 5.10.1. Max-Access: not-accessible Status: current	none
pnniNodePglEntry	`PnniNodePglEntry`	Contains the PGL election information of a PNNI logical node. Refer to *ATM Forum PNNI 1.0*, Section 5.10.1. Max-Access: not-accessible Status: current	none

Table 5-12 PGL Election Table MIB Objects (continued)

Name	Syntax	Description	Default Value
pnniNodePglLeadership Priority	`INTEGER (0..205)`	Specifies the leadership priority value the node can advertise in its nodal information group for the peer group. Only the value 0 is used with nodes that are not PGL or LGN capable. If there is no configured parent node index or no corresponding entry in the `pnniNodeTable`, the advertised leadership priority is 0 regardless of the value. Refer to *ATM Forum PNNI 1.0*, Section 5.10.1.2. Max-Access: read-create Status: current	`0`
pnniNodeCfgParentNode Index	`PnniNodeIndex`	If the node becomes peer group leader, the local node index is used to identify the node that will represent this peer group at the next level of hierarchy. The value 0 indicates there is no parent node. Refer to *ATM Forum PNNI 1.0*, Annex F. Max-Access: read-create Status: current	`0`
pnniNodePglInitTime	`Integer32`	The time, in seconds, this node can delay advertising its choice of the preferred PGL after having initialized operation, and reached the full state with at least one neighbor in the peer group. Refer to *ATM Forum PNNI 1.0*, Annex G, *PGLInitTime*. Units: seconds Max-Access: read-create Status: current	`15`
pnniNodePglOverrideDelay	`Integer32`	The time, in seconds that, a node waits to be declared the preferred PGL by unanimous agreement among its peers. In the absence of unanimous agreement, this is the amount of time that can pass before this node considers a two-thirds majority as a sufficient agreement. To declare itself a peer group leader, this node abandons the attempt to get an unanimous agreement. Refer to *ATM Forum PNNI 1.0*, Annex G, *Override Delay*. Units: seconds Max-Access: read-create Status: current	`30`

Table 5-12 PGL Election Table MIB Objects (continued)

Name	Syntax	Description	Default Value
pnniNodePglReelectTime	`Integer32`	The amount of time, in seconds that passes before, a node loses connectivity to the current PGL. This node can wait before restarting the process of electing a new PGL. Refer to *ATM Forum PNNI 1.0*, Annex G, *ReElectionInterval*. Units: seconds Max-Access: read-create Status: current	`15`
pnniNodePglState	`INTEGER { starting(1), awaiting (2), awaitingFull (3), initialDelay (4), calculating (5), awaitUnanimity (6), operPgl (7), operNotPgl (8), hungElection (9), awaitReElection(10)}`	Indicates the node state after the PGL election in the node peer group. The values are enumerated in the Peer Group Leader State Machine. Refer to *ATM Forum PNNI 1*, Section 5.10.1.1.2. Max-Access: read-only Status: current	none
pnniNodePreferredPgl	`PnniNodeId`	Specifies the local node ID for the peer group leader. This is also the value the local node is currently advertising in the `Preferred Peer Group Leader Node ID` field of its nodal information group within the given peer group. If a preferred PGL has not been chosen, this attribute value is set to (all) zero(s). Refer to *ATM Forum PNNI 1.0*, Section 5.10.1.1.6. Max-Access: read-only Status: current	none
pnniNodePeerGroupLeader	`PnniNodeId`	Specifies the Node Identifier that is operating as a PGL. If a PGL has not been elected, this attribute's value is set to (all) zero(s). Max-Access: read-only Status: current	none
pnniNodePglTimeStamp	`TimeStamp`	Establishes the time for the current PGL. Max-Access: read-only Status: current	none
pnniNodeActiveParentNodeId	`PnniNodeId`	Specifies the Node Identifier value used by the PGL to represent this peer group at the next level of the hierarchy. If this node is at the highest level of the hierarchy or if no PGL is elected, the PNNI Protocol Entity sets the value of this attribute to (all) zero(s). Max-Access: read-only Status: current	none

Initial Timer Values Table

The object identifiers for each MIB object are listed in Table 5-13.

Table 5-13 Initial Timer Values Table Object Identifiers

Name	Object Identifier
pnniNodeTimerTable	::= { pnniMIBObjects 4 }
pnniNodeTimerEntry	::= { pnniNodeTimerTable 1 }
pnniNodePtseHolddown	::= { pnniNodeTimerEntry 1 }
pnniNodeHelloHolddown	::= { pnniNodeTimerEntry 2 }
pnniNodeHelloInterval	::= { pnniNodeTimerEntry 3 }
pnniNodeHelloInactivityFactor	::= { pnniNodeTimerEntry 4 }
pnniNodeHlinkInact	::= { pnniNodeTimerEntry 5 }
pnniNodePtseRefreshInterval	::= { pnniNodeTimerEntry 6 }
pnniNodePtseLifetimeFactor	::= { pnniNodeTimerEntry 7 }
pnniNodeRxmtInterval	::= { pnniNodeTimerEntry 8 }
pnniNodePeerDelayedAckInterval	::= { pnniNodeTimerEntry 9 }
pnniNodeAvcrPm	::= { pnniNodeTimerEntry 10 }
pnniNodeAvcrMt	::= { pnniNodeTimerEntry 11 }
pnniNodeCdvPm	::= { pnniNodeTimerEntry 12 }
pnniNodeCtdPm	::= { pnniNodeTimerEntry 13 }

The MIB objects are listed in Table 5-14.

Table 5-14 Initial Timer Values Table MIB Objects

Name	Syntax	Description	Default Value
pnniNodeTimerTable	`Sequence of PnniNodeTimerEntry`	Contains a table of initial PNNI timer values and significant change thresholds. Max-Access: not- accessible Status: current	none
pnniNodeTimerEntry	`PnniNodeTimerEntry`	Contains an entry in the table for the initial PNNI timer values and significant change thresholds of a PNNI logical node. Max-Access: not-accessible Status: current	none

Table 5-14 Initial Timer Values Table MIB Objects (continued)

Name	Syntax	Description	Default Value
pnniNodePtseHolddown	Integer32	Specifies the initial value for the PNNI Topology State Element (PTSE) hold-down timer that is used by the node to limit the rate at which it can reoriginate PTSEs. It must be a positive, nonzero number. Refer to *ATM Forum PNNI 1.0*, Annex G, *MinPTSEInterval*. Units: 100 milliseconds Max-Access: read-create Status: current	10
pnniNodeHelloHolddown	Integer32	Specifies the initial value for the Hello hold-down timer that is used by the node to limit the rate it sends Hellos. It must be a positive nonzero number. Refer to the *ATM Forum PNNI 1.0*, Annex G, *MinHelloInterval*. Units: 100 ms Max-Access: read-create Status: current	10
pnniNodeHelloInterval	Integer32	Specifies the initial value for the Hello Timer. In the absence of triggered Hellos, this node sends one Hello packet on each of its ports. Refer to *ATM Forum PNNI 1.0*, Annex G, *HelloInterval*. Units: seconds Max-Access: read-create Status: current	15
pnniNodeHelloInactivityFactor	Integer32	Specifies the value for the Hello Inactivity factor that the node uses to determine when a neighbor has gone down. Refer to *ATM Forum PNNI 1.0*, Annex G, *InactivityFactor.* Max-Access: read-create Status: current	5
pnniNodeHlinkInact	Integer32	Specifies the time a node continues to advertise a horizontal (logical) link that it has not received and processed a Logical Group Node (LGN) Horizontal Link information group. Refer to *ATM Forum PNNI 1.0*, Annex G, *HorizontalLinkInactivityTime*. Units: seconds Max-Access: read-create Status: current	120

Table 5-14 Initial Timer Values Table MIB Objects (continued)

Name	Syntax	Description	Default Value
pnniNodePtseRefreshInterval	Integer32	Specifies the initial value for the refresh timer that this node uses to drive origination of PTSEs in the absence of triggered updates. Refer to *ATM Forum PNNI 1.0*, Annex G, *PTSERefreshInterval*. Units: seconds Max-Access: read-create Status: current	1800
pnniNodePtseLifetimeFactor	INTEGER (101..1000)	Specifies the value for the lifetime multiplier that is expressed as a percentage. The result of multiplying the pnniNodePtseRefreshInterval attribute value is used as the initial lifetime for the self-originated PTSEs. Refer to *ATM Forum PNNI 1.0*, Annex G, *PTSELifetimeFactor*. Units: percent Max-Access: read-create Status: current	200
pnniNodeRxmtInterval	Integer32	Specifies the time between retransmission of unacknowledged database summary packets, PTSE request packets, and PNNI Topology State Packets (PTSPs). Refer to *ATM Forum PNNI 1.0*, Annex G, *DSRxmtInterval*, *RequestRxmtInterval*, *PTSERetransmissionInterval*. Units: seconds Max-Access: read-create Status: current	5
pnniNodePeerDelayedAckInterval	Integer32	Specifies the minimum time between transmissions of delayed PTSE acknowledgement packets. Refer to *ATM Forum PNNI 1.0*, Annex G, *PeerDelayedAckInterval*, Appendix G. Units: 100 ms Max-Access: read-create Status: current	10
pnniNodeAvcrPm	INTEGER (1..99)	Specifies the proportional multiplier used in the algorithms. The multiplier determines a significant change for AvCR parameters, which are expressed as a percentage. Refer to *ATM Forum PNNI 1.0*, Section 5.8.5.2.5.4, Annex G, *AvCR_PM*. Units: percent Max-Access: read-create Status: current	50

Table 5-14 Initial Timer Values Table MIB Objects (continued)

Name	Syntax	Description	Default Value
pnniNodeAvcrMt	INTEGER (1..99)	Specifies the minimum threshold used in the algorithms. The threshold determines a significant change for AvCR parameters, which are expressed as a percentage. Refer to *ATM Forum PNNI 1.0*, Section 5.8.5.2.5.4, Annex G, *AvCR_mT*. Units: percent Max-Access: read-create Status: current	3
pnniNodeCdvPm	INTEGER (1..99)	Specifies the proportional multiplier used in the algorithms. The multiplier determines significant change for Cell Delay Variation (CDV) metrics, which are expressed as a percentage. Refer to *ATM Forum PNNI 1.0*, Section 5.8.5.2.5.6, Annex G, *CDV_PM*. Units: percent Max-Access: read-create Status: current	25
pnniNodeCtdPm	INTEGER (1..99)	Specifies the proportional multiplier used in the algorithms. The multiplier determines a significant change for Cell Transfer Delay (CTD) metrics, which are expressed as a percentage. Refer to *ATM Forum PNNI 1.0*, Section 5.8.5.2.5.5, Annex G, *maxCTD_PM*. Units: percent Max-Access: read-create Status: current	50

Nodal SVCC-Based RCC Variables Table

The object identifiers for each MIB object is listed in Table 5-15.

Table 5-15 Nodal SVCC-Based RCC Variables Table Object Identifiers

Name	Object Identifier
pnniNodeSvccTable	::= { pnniMIBObjects 5 }
pnniNodeSvccEntry	::= { pnniNodeSvccTable 1 }
pnniNodeSvccInitTime	::= { pnniNodeSvccEntry 1 }
pnniNodeSvccRetryTime	::= { pnniNodeSvccEntry 2 }
pnniNodeSvccCallingIntegrityTime	::= { pnniNodeSvccEntry 3 }

Table 5-15 Nodal SVCC-Based RCC Variables Table Object Identifiers (continued)

Name	Object Identifier
pnniNodeSvccCalledIntegrityTime	::= { pnniNodeSvccEntry 4 }
pnniNodeSvccTrafficDescriptorIndex	::= { pnniNodeSvccEntry 5 }

The MIB objects are listed in Table 5-16.

Table 5-16 Nodal SVCC-Based RCC Variables Table MIB Objects

Name	Syntax	Description	Default Value
pnniNodeSvccTable	`Sequence of PnniNodeSvccEntry`	Provides a table of variables related to SVCC-based routing control channels. Refer to *ATM Forum PNNI 1.0*, Section 5.5. Max-Access: not-accessible Status: current	none
pnniNodeSvccEntry	`PnniNodeSvccEntry`	Specifies a table entry that contains SVCC-based RCC variables of a PNNI logical node. Refer to *ATM Forum PNNI 1.0*, Section 5.5. Max-Access: not-accessible Status: current	none
pnniNodeSvccInitTime	`Integer32`	Specifies the time that this node waits before initiating establishment of an SVCC to a neighbor with a numerically lower ATM address. This happens after determining that a SVCC can be established. Refer to *ATM Forum PNNI 1.0*, Annex G, *InitialLGNSVCTimeout*. Units: seconds Max-Access: read-create Status: current	`4`
pnniNodeSvccRetryTime	`Integer32`	Specifies the amount of time this node waits before attempting to reestablish connection with a viable SVCC-based RCC. Refer to *ATM Forum PNNI 1.0*, Annex G, *RetryLGNSVCTimeout*. Units: seconds Max-Access: read-create Status: current	`30`

Table 5-16 Nodal SVCC-Based RCC Variables Table MIB Objects (continued)

Name	Syntax	Description	Default Value
pnniNodeSvccCalling IntegrityTime	`Integer32`	Specifies the time this node waits for SVCC. The node initiates an establishment to the calling party to become established before it gives up. Refer to *ATM Forum PNNI 1.0*, Annex G, *SVCCallingIntegrityTime*. Units: seconds Max-Access: read-create Status: current	`35`
pnniNodeSvccCalled IntegrityTime	`Integer32`	Specifies the time this node will wait for SVCC. The node decides to accept the called party to become established before it gives up. Refer to *ATM Forum PNNI 1.0,* Annex G, *SVCCalledIntegrityTime*. Units: seconds Max-Access: read-create Status: current	`50`
pnniNodeSvccTraffic DescriptorIndex	`AtmTrafficDescrParamIndex`	Specifies the index for the `atmTrafficDescrParamTable` defined in RFC 1695. This traffic descriptor is used when establishing switched virtual channels for SVCC-based RCCs to the PNNI logical group nodes. Refer to *ATM Forum PNNI 1.0*,Section 5.5.2, Annex G, *RCCMaximumBurstSize*, *RCCPeakCellRate*, *RCCSustainableCellRate*. Max-Access: read-create Status: current	none

Scope Mapping Table

The object identifier for each MIB object is listed in Table 5-17.

Table 5-17 Scope Mapping Table Object Identifiers

Name	Object Identifier
pnniScopeMappingTable	::= { pnniMIBObjects 6 }
pnniScopeMappingEntry	::= { pnniScopeMappingTable 1 }
pnniScopeLocalNetwork	::= { pnniScopeMappingEntry 1 }
pnniScopeLocalNetworkPlusOne	::= { pnniScopeMappingEntry 2 }
pnniScopeLocalNetworkPlusTwo	::= { pnniScopeMappingEntry 3 }
pnniScopeSiteMinusOne	::= { pnniScopeMappingEntry 4 }
pnniScopeIntraSite	::= { pnniScopeMappingEntry 5 }

Table 5-17 Scope Mapping Table Object Identifiers (continued)

Name	Object Identifier
pnniScopeSitePlusOne	::= { pnniScopeMappingEntry 6 }
pnniScopeOrganizationMinusOne	::= { pnniScopeMappingEntry 7 }
pnniScopeIntraOrganization	::= { pnniScopeMappingEntry 8 }
pnniScopeOrganizationPlusOne	::= { pnniScopeMappingEntry 9 }
pnniScopeCommunityMinusOne	::= { pnniScopeMappingEntry 10 }
pnniScopeIntraCommunity	::= { pnniScopeMappingEntry 11 }
pnniScopeCommunityPlusOne	::= { pnniScopeMappingEntry 12 }
pnniScopeRegional	::= { pnniScopeMappingEntry 13 }
pnniScopeInterRegional	::= { pnniScopeMappingEntry 14 }
pnniScopeGlobal	::= { pnniScopeMappingEntry 15 }

A list of the MIB objects is described in Table 5-18.

Table 5-18 Scope Mapping Table MIB Objects

Name	Syntax	Description	Default Value
pnniScopeMappingTable	`Sequence of PnniScopeMappingEntry`	Contains the mappings of membership and connection scope from organizational scope values (used at UNI interfaces) to PNNI scope. For example, PNNI routing level indicators. Refer to *ATM Forum PNNI 1.0*, Section 5.3.6. Max-Access: not-accessible Status: current	none
pnniScopeMappingEntry	`PnniScopeMappingEntry`	Contains the scope mapping information for a PNNI logical node. Refer to *ATM Forum PNNI 1.0*, Section 5.3.6. Max-Access: not-accessible Status: current	none
pnniScopeLocalNetwork	`PnniLevel`	Specifies the highest level of the PNNI hierarchy, for example, the smallest PNNI routing level within the organizational scope value `localNetwork(1)`. Max-Access: read-create Status: current	`96`
pnniScopeLocalNetworkPlusOne	`PnniLevel`	Specifies the highest level of the PNNI hierarchy, for example, the smallest PNNI routing level within the organizational scope value `localNetworkPlusOne(2)`. Max-Access: read-create Status: current	`96`

Table 5-18 Scope Mapping Table MIB Objects (continued)

Name	Syntax	Description	Default Value
pnniScopeLocalNetworkPlusTwo	PnniLevel	Specifies the highest level of the PNNI hierarchy, for example, the smallest PNNI routing level within the organizational scope value localNetworkPlusTwo(3). Max-Access: read-create Status: current	96
pnniScopeSiteMinusOne	PnniLevel	Specifies the highest level of the PNNI hierarchy, for example, the smallest PNNI routing level within the organizational scope value siteMinusOne(4). Max-Access: read-create Status: current	80
pnniScopeIntraSite	PnniLevel	Specifies the highest level of the PNNI hierarchy, for example, the smallest PNNI routing level within the organizational scope value intraSite(5). Max-Access: read-create Status: current	80
pnniScopeSitePlusOne	PnniLevel	Specifies the highest level of the PNNI hierarchy, for example, the smallest PNNI routing level within the organizational scope value sitePlusOne(6). Max-Access: read-create Status: current	72
pnniScopeOrganizationMinusOne	PnniLevel	Specifies the highest level of the PNNI hierarchy, for example, the smallest PNNI routing level within the organizational scope value organizationMinusOne(7). Max-Access: read-create Status: current	72
pnniScopeIntraOrganization	PnniLevel	Specifies the highest level of the PNNI hierarchy, for example, the smallest PNNI routing level within the organizational scope value intraOrganization(8). Max-Access: read-create Status: current	64
pnniScopeOrganizationPlusOne	PnniLevel	Specifies the highest level of the PNNI hierarchy, for example, the smallest PNNI routing level within the organizational scope value organizationPlusOne(9). Max-Access: read-create Status: current	64

Table 5-18 Scope Mapping Table MIB Objects (continued)

Name	Syntax	Description	Default Value
pnniScopeCommunityMinusOne	PnniLevel	Specifies the highest level of PNNI hierarchy, for example, the smallest PNNI routing level within the organizational scope value communityMinusOne(10). Max-Access: read-create Status: current	64
pnniScopeIntraCommunity	PnniLevel	Specifies the highest level of PNNI hierarchy, for example, the smallest PNNI routing level within the organizational scope value intraCommunity(11). Max-Access: read-create Status: current	48
pnniScopeCommunityPlusOne	PnniLevel	Specifies the highest level of the PNNI hierarchy, for example, the smallest PNNI routing level within the organizational scope value communityPlusOne(12). Max-Access: read-create Status: current	48
pnniScopeRegional	PnniLevel	Specifies the highest level of the PNNI hierarchy, for example, the smallest PNNI routing level within the organizational scope value regional(13). Max-Access: read-create Status: current	32
pnniScopeInterRegional	PnniLevel	Specifies the highest level of the PNNI hierarchy, for example, the smallest PNNI routing level within the organizational scope value interRegional(14). Max-Access: read-create Status: current	32
pnniScopeGlobal	PnniLevel	Specifies the highest level of the PNNI hierarchy, for example, the smallest PNNI routing level within the organizational scope value global(15). Max-Access: read-create Status: current	0

Summary Address Table

The object identifier for each MIB object is listed in Table 5-19.

Table 5-19 Summary Address Table Object Identifiers

Name	Object Identifier
pnniSummaryAddressTable	::= { pnniMIBObjects 20 }
pnniSummaryAddressEntry	::= { pnniSummaryAddressTable 1 }
pnniSummaryAddressType	::= { pnniSummaryAddressEntry 1 }
pnniSummaryAddressAddress	::= { pnniSummaryAddressEntry 2 }
pnniSummaryAddressPrefixLength	::= { pnniSummaryAddressEntry 3 }
pnniSummaryAddressSuppress	::= { pnniSummaryAddressEntry 4 }
pnniSummaryAddressState	::= { pnniSummaryAddressEntry 5 }
pnniSummaryAddressRowStatus	::= { pnniSummaryAddressEntry 6 }

The MIB objects are listed in Table 5-20.

Table 5-20 Summary Address Table MIB Objects

Name	Syntax	Description	Default Value
pnniSummaryAddress Table	`Sequence of PnniSummaryAddressEntry`	Specifies a list of the summary address prefixes that can be advertised by the logical PNNI entity. Refer to *ATM Forum PNNI 1.0*, Section 5.9.2. Max-Access: not-accessible Status: current	none
pnniSummaryAddress Entry	`PnniSummaryAddressEntry`	Specifies an entry in the table that contains the summary address prefix information. Refer to *ATM Forum PNNI 1.0*, Section 5.9.2. Max-Access: not-accessible Status: current The following list the indexes: • `pnniNodeIndex` • `pnniSummaryAddressType` • `pnniSummaryAddressAddress` • `pnniSummaryAddressPrefixLength`	none
pnniSummaryAddress Type	`Integer {internal (1), exterior (2)}`	Describes the type of summary, for example, internal or exterior. Max-Access: not-accessible Status: current	none

Table 5-20 Summary Address Table MIB Objects (continued)

Name	Syntax	Description	Default Value
pnniSummaryAddress Address	`AtmAddrPrefix`	Describes the ATM End System Address prefix for the summary. Max-Access: not-accessible Status: current	none
pnniSummaryAddress PrefixLength	`PnniPrefixLength`	Determines the prefix length for the summary. Max-Access: not-accessible Status: current	none
pnniSummaryAddress Suppress	`TruthValue`	Determines the addresses that are being summarized by the instance. The default value `false` indicates that the summary can propagate into the peer group. Network Management is able to set the value of this attribute to suppress `true`. This attribute suppresses the summary and any reachable addresses it summarizes from being advertised into the peer group. Max-Access: read-create Status: current	`false`
pnniSummaryAddress State	`Integer {advertising (1), suppressing (2), inactive (3)}`	Indicates whether the summary is advertised by the node within the local switching system into its peer group. Max-Access: read-only Status: current	none
pnniSummaryAddress RowStatus	`RowStatus`	Creates, deletes, activates, and deactivates a summary. Max-Access: read-create Status: current	none

Link Table

The object identifier for each MIB object is listed in Table 5-21.

Table 5-21 Link Table Object Identifiers

Name	Object Identifier
pnniLinkTable	::= { pnniMIBObjects 9 }
pnniLinkEntry	::= { pnniLinkTable 1 }
pnniLinkPortId	::= { pnniLinkEntry 1 }
pnniLinkType	::= { pnniLinkEntry 2 }
pnniLinkVersion	::= { pnniLinkEntry 3 }
pnniLinkHelloState	::= { pnniLinkEntry 4 }
pnniLinkRemoteNodeId	::= { pnniLinkEntry 5 }
pnniLinkRemotePortId	::= { pnniLinkEntry 6 }

Table 5-21 Link Table Object Identifiers (continued)

Name	Object Identifier
pnniLinkDerivedAggrToken	::= { pnniLinkEntry 7 }
pnniLinkUpnodeId	::= { pnniLinkEntry 8 }
pnniLinkUpnodeAtmAddress	::= { pnniLinkEntry 9 }
pnniLinkCommonPeerGroupId	::= { pnniLinkEntry 10 }
pnniLinkIfIndex	::= { pnniLinkEntry 11 }
pnniLinkSvccRccIndex	::= { pnniLinkEntry 12 }
pnniLinkRcvHellos	::= { pnniLinkEntry 13 }
pnniLinkXmtHellos	::= { pnniLinkEntry 14 }

The MIB objects are listed in Table 5-22.

Table 5-22 Link Table MIB Objects

Name	Syntax	Description	Default Value
pnniLinkTable	`Sequence of PnniLinkEntry`	Contains the attributes to describe the operation of logical links attached to the local switching system and the relationship with the neighbor nodes on the other end of the links. Links are attached to a specific node within the switching system. A concatenation of the node index within the local switching system and the port ID are used as the instance ID to uniquely identify the link. Links can represent horizontal links between lowest level neighboring peers, outside links, uplinks, or horizontal links to LGN. The entire `pnniLink` object is read-only, reflecting the fact that this information is discovered dynamically by the PNNI protocol rather than configured. Refer to *ATM Forum PNNI 1.0*, Section 5.6. Max-Access: not accessible Status: current	none
pnniLinkEntry	`PnniLinkEntry`	Specifies an entry in the table that contains information about a link attached to a PNNI logical node. Refer to *ATM Forum PNNI 1.0*, Section 5.6. Max-Access: not-accessible Status: current The following list the indexes: • `pnniNodeIndex` • `pnniLinkPortId`	none

Table 5-22 Link Table MIB Objects (continued)

Name	Syntax	Description	Default Value
pnniLinkPortId	PnniPortId	Contains the port identifier of the link that is selected by the local node. This value is only within the context of the node to the attached port. Max-Access: not-accessible Status: current	none
pnniLinkType	INTEGER {unknown (1), lowestLevelHorizontal Link (2), horizontalLinkToFrom Lgn (3), lowestLevelOutside Link (4), uplink (5), outsideLinkAndUp link (6)}	Describes the type of link. Max-Access: read-only Status: current	none
pnniLinkVersion	PnniVersion	Indicates the version of the PNNI routing protocol used to exchange information over the link for horizontal and outside links between low-level nodes, and for links of an unknown type. If communication with the neighbor node is not established, version is set to unknown. For uplinks (where the port ID is not also used for the underlying outside link) or links to LGN, the version is set to unknown. Max-Access: read-only Status: current	none
pnniLinkHelloState	PnniHelloState	Indicates the state of the Hello protocol exchange over the link for horizontal and outside links between low-level nodes and for links of an unknown type. For links to LGN, this attribute indicates the state of the corresponding LGN Horizontal Link Hello State Machine. For uplinks (where the port ID is not also used for the underlying outside link), this attribute is set to notApplicable. Refer to *ATM Forum PNNI 1.0*, Section 5.6.2.1.2. Max-Access: read-only Status: current	none
pnniLinkRemoteNodeId	PnniNodeId	Indicates the node identifier of the remote (neighboring) node on the other end of the link. If the pnniLinkType is outside link and uplink, this is the node identifier of the lowest-level neighbor node on the other end of the outside link. If the remote node ID is unknown or if the pnniLinkType is uplink, this attribute is set to all zeros. Max-Access: read-only Status: current	none

Table 5-22 Link Table MIB Objects (continued)

Name	Syntax	Description	Default Value
pnniLinkRemotePortId	PnniPortId	Indicates the port identifier of the port at the remote end of the link as assigned by the remote node. If the pnniLinkType is outside link and uplink, this is the port identifier assigned by the lowest-level neighbor node to identify the outside link. If the remote port ID is unknown or if the pnniLinkType is uplink, this attribute is set to 0. Max-Access: read-only Status: current	none
pnniLinkDerivedAggr Token	PnniAggrToken	Indicates the derived aggregation token value used on this link. For horizontal links between lowest-level nodes and when the link type is not yet known, this attribute takes the value of 0. Refer to *ATM Forum PNNI 1.0*, Section 5.10.3.1. Max-Access: read-only Status: current	none
pnniLinkUpnodeId	PnniNodeId	Contains the node identifier of the upnode (the neighbor node identity at the level of the common peer group) for outside links and uplinks. When the upnode has not yet been identified, this attribute is set to 0. For horizontal links or the link type is not known, this attribute is set to 0. Max-Access: read-only Status: current	none
pnniLinkUpnodeAtm Address	PnniAtmAddr	Contains the ATM end system address used to establish connections to the upnode for outside links and uplinks. When the upnode is not identified, this attribute is set to 0. For horizontal links or the link type is not known, this attribute is set to 0. Max-Access: read-only Status: current	none
pnniLinkCommonPeer GroupId	PnniPeerGroupId	Contains the peer group identifier of the low-level common peer group in the ancestry of the neighboring node, and the node within the local switching system for outside links and uplinks. The value of this attribute takes on a value determined by the Hello exchange of hierarchical information that occurs between the two low-level border nodes. When the common peer group is not identified, this attribute is set to 0. For horizontal links or when the link type is not known, this attribute is set to all zeros. Max-Access: read-only Status: current	none

Table 5-22 Link Table MIB Objects (continued)

Name	Syntax	Description	Default Value
pnniLinkIfIndex	`InterfaceIndex`	Identifies the interface that the logical link corresponds for horizontal and outside links between low-level nodes and for links of an unknown type. For all other cases, the value of this object is 0. Max-Access: read-only Status: current	none
pnniLinkSvccRccIndex	`PnniSvccRccIndex`	Identifies the SVCC-based RCC used to exchange information with the neighboring peer logical group node for horizontal links to LGN. If the `pnniLinkType` is not a horizontal link to LGN, this attribute is set to the value of 0. Max-Access: read-only Status: current	none
pnniLinkRcvHellos	`Counter32`	Contains a count of the number of hello packets received over this link for horizontal and outside links between low-level nodes and for links of an unknown type. If `pnniLinkType` is a horizontal link to LGN or uplink, this attribute is set to the value of 0. Max-Access: read-only Status: current	none
pnniLinkXmtHellos	`Counter32`	Contains a count of the number of Hello packets transmitted over this link for horizontal and outside links between low-level nodes and for links of an unknown type. If `pnniLinkType` is a horizontal link to LGN or uplink, this attribute is set to the value of 0. Max-Access: read-only Status: current	none

ATM PNNI MIB Conformance and Compliance Statements

The information on conformance is specific to SNMPv2. Table 5-23 lists the conformance groups.

Table 5-23 ATM PNNI MIB Conformance Groups

Name	Object Identifier
pnniMIBConformance	::= { pnniMIB 2 }
pnniMIBCompliances	::= { pnniMIBConformance 1 }
pnniMIBGroups	::= { pnniMIBConformance 2 }

ATM PNNI MIB Compliance Statements

The object identifier used for `pnniIfVPCapability` is listed in Table 5-24.

Table 5-24 ATM PNNI MIB Compliance Statement

Name	Object Identifier
pnniIfVPCapability	::= { pnniMIBCompliances 1 }

The compliance statements used for each object are listed in Table 5-25.

Table 5-25 ATM PNNI MIB Objects Used for Compliance Statements

Name	Syntax	Description	Default Value
pnniMIBCompliance (module compliance)	none	Specifies the compliance statement for entities that implement the PNNI MIB. The following nodes are used for management of PNNI objects: • Minimum function node is identified by the suffix `MinGroup`. • Border node is identified by the suffix `BorderGroup`. • PGL or LGN capable node is identified by the suffix `LgnGroup`. The group of optional PNNI objects is identified by the suffix `OptionalGroup`. Status: current This module contains the following mandatory groups: • `pnniGeneralMinGroup` • `pnniNodeMinGroup` • `pnniNodePglMinGroup` • `nniNodeTimerMinGroup` • `pnniScopeMinGroup` • `pnniIfMinGroup` • `pnniLinkMinGroup` • `pnniNbrPeerMinGroup` • `pnniNbrPeerPortMinGroup`	none
pnniNodeId	none	Supports a manual configuration of the node ID that is optional. Min-Access: read-only	none
pnniNodeLowest	none	Provides write or creates access to the `pnniNodeLowest` object that is PGL LGN capable. Min-Access: read-only	none

Table 5-25 ATM PNNI MIB Objects Used for Compliance Statements (continued)

Name	Syntax	Description	Default Value
pnniNodeRestrictedTransit	none	Supports the restricted transit capability that is optional. Min-Access: read-only	none
pnniNodeComplexRep	none	Generates the complex node representation that is required only for PGL or LGN capable switching systems, which is optional. Min-Access: read-only	none
pnniNodeRowStatus	`INTEGER { active(1) }`	Creates more than one node in a switching system that is optional. Min-Access: read-only	none
pnniNodePglLeadershipPriority	none	Provides write or create access to the `pnniNodePglLeadershipPriority` object that is PGL or LGN capable. Min-Access: read-only	none
pnniIfNodeIndex	none	Provides write access to the `pnniIfNodeIndex` object that is optional. This object applies only when there are multiple lowest-level nodes. Min-Access: read-only	none
pnniIfVPCapability	none	Supports switched virtual paths that are optional. Min-Access: read-only	none

ATM PNNI MIB Units of Conformance Statements

The object identifier for each MIB object is listed in Table 5-26.

Table 5-26 ATM PNNI MIB Units of Conformance Object Identifiers

Name	Object Identifier
pnniGeneralMinGroup	::= { pnniMIBGroups 1 }
pnniGeneralBorderGroup	::= { pnniMIBGroups 2 }
pnniNodeMinGroup	::= { pnniMIBGroups 3 }
pnniNodePglMinGroup	::= { pnniMIBGroups 4 }
pnniNodePglLgnGroup	::= { pnniMIBGroups 5 }
pnniNodeTimerMinGroup	::= { pnniMIBGroups 6 }
pnniNodeTimerLgnGroup	::= { pnniMIBGroups 7 }
pnniNodeSvccLgnGroup	::= { pnniMIBGroups 8 }
pnniScopeMinGroup	::= { pnniMIBGroups 9 }
pnniSummaryAddressLgnGroup	::= { pnniMIBGroups 31 }
pnniIfMinGroup	::= { pnniMIBGroups 11 }

Table 5-26 ATM PNNI MIB Units of Conformance Object Identifiers (continued)

Name	Object Identifier
pnniIfBorderGroup	::= { pnniMIBGroups 12 }
pnniLinkMinGroup	::= { pnniMIBGroups 13 }
pnniLinkBorderOrLgnGroup	::= { pnniMIBGroups 14 }
pnniLinkLgnGroup	::= { pnniMIBGroups 15 }
pnniNbrPeerMinGroup	::= { pnniMIBGroups 16 }
pnniNbrPeerLgnGroup	::= { pnniMIBGroups 17 }
pnniNbrPeerPortMinGroup	::= { pnniMIBGroups 18 }
pnniSvccRccLgnGroup	::= { pnniMIBGroups 19 }
pnniPtseOptionalGroup	::= { pnniMIBGroups 20 }
pnniMapOptionalGroup	::= { pnniMIBGroups 21 }
pnniMapNodeOptionalGroup	::= { pnniMIBGroups 22 }
pnniMapAddrOptionalGroup	::= { pnniMIBGroups 23 }
pnniMapTnsOptionalGroup	::= { pnniMIBGroups 24 }
pnniMetricsOptionalGroup	::= { pnniMIBGroups 25 }
pnniRouteGeneralOptionalGroup	::= { pnniMIBGroups 26 }
pnniRouteNodeOptionalGroup	::= { pnniMIBGroups 27 }
pnniDTLOptionalGroup	::= { pnniMIBGroups 28 }
pnniRouteAddrOptionalGroup	::= { pnniMIBGroups 29 }
pnniRouteTnsOptionalGroup	::= { pnniMIBGroups 30 }

Table 5-27 describes the objects used for units of conformance.

Table 5-27 ATM PNNI MIB Objects Used for Units of Conformance

Name	Objects	Description	Default Value
pnniGeneralMinGroup	`pnniHighestVersion`, `pnniLowestVersion`, `pnniDtlCountOriginator`, `pnniCrankbackCountOriginator`, `pnniAltRouteCountOriginator`, `pnniRouteFailCountOriginator`, `pnniRouteFailUnreachable Originator`	Specifies a collection of general PNNI objects required for management of a minimum function switching system. Status: current	none
pnniGeneralBorder Group	`pnniDtlCountBorder`, `pnniCrankbackCountBorder`, `pnniAltRouteCountBorder`, `pnniRouteFailCountBorder`, `pnniRouteFailUnreachableBorder`	Specifies a collection of general PNNI objects required for management of a border node. Status: current	none

Table 5-27 ATM PNNI MIB Objects Used for Units of Conformance (continued)

Name	Objects	Description	Default Value
pnniNodeMinGroup	pnniNodeLevel, pnniNodeId, pnniNodeLowest, pnniNodeAdminStatus, pnniNodeOperStatus, pnniNodeAtmAddress, pnniNodePeerGroupId, pnniNodeRestrictedTransit, pnniNodeComplexRep, pnniNodeRestrictedBranching, pnniNodeDatabaseOverload, pnniNodePtses, pnniNodeRowStatus	Specifies a collection of per-node PNNI objects required for management of a minimum functional switching system. Status: current	none
pnniNodePglMin Group	pnniNodePglLeadershipPriority, pnniNodePglInitTime, pnniNodePglReelectTime, pnniNodePglState, pnniNodePreferredPgl, pnniNodePeerGroupLeader, pnniNodePglTimeStamp, pnniNodeActiveParentNodeId	Specifies a collection of per-node, PGL election-related PNNI objects required for managing a minimum functional switching system. Status: current	none
pnniNodePglLgn Group	pnniNodeCfgParentNodeIndex, pnniNodePglOverrideDelay	Specifies a collection of per-node, PGL election-related PNNI objects required for managing a PGL or LGN capable switching system. Status: current	none
pnniNodeTimerMin Group	pnniNodePtseHolddown, pnniNodeHelloHolddown, pnniNodeHelloInterval, pnniNodeHelloInactivityFactor, pnniNodePtseRefreshInterval, pnniNodePtseLifetimeFactor, pnniNodeRxmtInterval, pnniNodePeerDelayedAckInterval, pnniNodeAvcrPm, pnniNodeAvcrMt, pnniNodeCdvPm, pnniNodeCtdPm	Specifies a collection of per-node PNNI objects required for managing timers and significant change thresholds in a minimum functional switching system. Status: current	none
pnniNodeTimerLgn Group	pnniNodeHlinkInact	Specifies a collection of per-node PNNI objects required for managing timers in a PGL or LGN capable switching system. Status: current	none
pnniNodeSvccLgn Group	pnniNodeSvccInitTime, pnniNodeSvccRetryTime, pnniNodeSvccCallingIntegrityTime, pnniNodeSvccCalledIntegrityTime, pnniNodeSvccTrafficDescriptorIndex	Specifies a collection of per-node, SVCC-based RCC related PNNI objects required for managing a PGL or LGN capable switching system. Status: current	none

Table 5-27 ATM PNNI MIB Objects Used for Units of Conformance (continued)

Name	Objects	Description	Default Value
pnniScopeMinGroup	pnniScopeLocalNetwork, pnniScopeLocalNetworkPlusOne, pnniScopeLocalNetworkPlusTwo, pnniScopeSiteMinusOne, pnniScopeIntraSite, pnniScopeSitePlusOne, pnniScopeOrganizationMinusOne, pnniScopeIntraOrganization, pnniScopeOrganizationPlusOne, pnniScopeCommunityMinusOne, pnniScopeIntraCommunity, pnniScopeCommunityPlusOne, pnniScopeRegional, pnniScopeInterRegional, pnniScopeGlobal	Specifies a collection of per-node scope mapping-related PNNI objects required for managing a minimum functional switching system. Status: current	none
pnniSummaryAddressLgnGroup	pnniSummaryAddressSuppress, pnniSummaryAddressState, pnniSummaryAddressRowStatus	Specifies a collection of PNNI objects required for controlling address summarization. Status: current	none
pnniIfMinGroup	pnniIfNodeIndex, pnniIfPortId, pnniIfVPCapability, pnniIfAdmWeightCbr, pnniIfAdmWeightRtVbr, pnniIfAdmWeightNrtVbr, pnniIfAdmWeightAbr, pnniIfAdmWeightUbr, pnniIfRccServiceCategory, pnniIfRccTrafficDescrIndex	Specifies a collection of per-interface PNNI objects required for managing a minimum functional switching system. Status: current	none
pnniIfBorderGroup	pnniIfAggrToken	Specifies a collection of per-interface PNNI objects required for managing a border node. Status: current	none
pnniLinkMinGroup	pnniLinkType, pnniLinkVersion, pnniLinkHelloState, pnniLinkRemoteNodeId, pnniLinkRemotePortId, pnniLinkIfIndex, pnniLinkRcvHellos, pnniLinkXmtHellos	Specifies a collection of per-link PNNI objects required for managing a minimum functional switching system. Status: current	none
pnniLinkBorderOrLgnGroup	pnniLinkDerivedAggrToken, pnniLinkUpnodeId, pnniLinkUpnodeAtmAddress, pnniLinkCommonPeerGroupId	Specifies a collection of per-link PNNI objects required for managing a border node or a PGL or LGN capable switching system. Status: current	none
pnniLinkLgnGroup	pnniLinkSvccRccIndex	Specifies a collection of per-link PNNI objects required for managing a PGL or LGN capable switching system. Status: current	none

Table 5-27 ATM PNNI MIB Objects Used for Units of Conformance (continued)

Name	Objects	Description	Default Value
pnniNbrPeerMin Group	pnniNbrPeerState, pnniNbrPeerPortCount, pnniNbrPeerRcvDbSums, pnniNbrPeerXmtDbSums, pnniNbrPeerRcvPtsps, pnniNbrPeerXmtPtsps, pnniNbrPeerRcvPtseReqs, pnniNbrPeerXmtPtseReqs, pnniNbrPeerRcvPtseAcks, pnniNbrPeerXmtPtseAcks	Specifies a collection of per-neighboring-peer PNNI objects required for managing a minimum functional switching system. Status: current	none
pnniNbrPeerLgn Group	pnniNbrPeerSvccRccIndex	Specifies a collection of per-neighboring-peer PNNI objects required for managing a PGL or LGN capable switching system. Status: current	none
pnniNbrPeerPortMin Group	pnniNbrPeerPortFloodStatus	Specifies a collection of per-port-to-neighboring-peer PNNI objects required for managing a minimum functional switching system. Status: current	none
pnniSvccRccLgnGroup	pnniSvccRccVersion, pnniSvccRccHelloState, pnniSvccRccRemoteNodeId, pnniSvccRccRemoteAtmAddress, pnniSvccRccRcvHellos, pnniSvccRccXmtHellos, pnniSvccRccIfIndex, pnniSvccRccVpi, pnniSvccRccVci	Specifies a collection of per-SVCC-based RCC PNNI objects required for managing a PGL or LGN capable switching system. Status: current	none
pnniPtseOptional Group	pnniPtseType, pnniPtseSequenceNum, pnniPtseChecksum, pnniPtseLifeTime, pnniPtseInfo	Specifies a collection of the optional per PTSE PNNI objects. Status: current	none
pnniMapOptional Group	pnniMapType, pnniMapPeerGroupId, pnniMapAggrToken, pnniMapRemoteNodeId, pnniMapRemotePortId, pnniMapVPCapability, pnniMapPtseId, pnniMapMetricsTag	Specifies a collection of the optional PNNI objects used to create a map of nodes and links in the PNNI routing domain. Status: current	none

Table 5-27 ATM PNNI MIB Objects Used for Units of Conformance (continued)

Name	Objects	Description	Default Value
pnniMapNodeOptional Group	pnniMapNodePeerGroupId, pnniMapNodeAtmAddress, pnniMapNodeRestrictedTransit, pnniMapNodeComplexRep, pnniMapNodeRestrictedBranching, pnniMapNodeDatabaseOverload, pnniMapNodeIAmLeader, pnniMapNodeLeadershipPriority, pnniMapNodePreferredPgl, pnniMapNodeParentNodeId, pnniMapNodeParentAtmAddress, pnniMapNodeParentPeerGroupId, pnniMapNodeParentPglNodeId	Specifies a collection of the optional PNNI objects used to create a map of nodes in the PNNI routing domain. Status: current	none
pnniMapAddrOptional Group	pnniMapAddrAddress, pnniMapAddrPrefixLength	Specifies a collection of the optional PNNI objects used to create a map of reachable addresses in the PNNI routing domain. Status: current	none
pnniMapTnsOptional Group	pnniMapTnsId	Specifies a collection of the optional PNNI objects used to create a map of reachable transit networks in the PNNI routing domain. Status: current	none
pnniMetricsOptional Group	pnniMetricsClasses, pnniMetricsGcacClp, pnniMetricsAdminWeight, pnniMetrics1, pnniMetrics2, pnniMetrics3, pnniMetrics4, pnniMetrics5, pnniMetrics6, pnniMetrics7, pnniMetrics8, pnniMetricsRowStatus	Specifies a collection of the optional PNNI objects used to manage metrics and attributes associated with PNNI entities. Status: current	none
pnniRouteGeneral OptionalGroup	pnniRouteNodeNumber, pnniRouteAddrNumber	Specifies a collection of the optional PNNI objects that includes pnniRouteNodeNumber and pnniRouteAddrNumber. Status: current	none

Table 5-27 ATM PNNI MIB Objects Used for Units of Conformance (continued)

Name	Objects	Description	Default Value
pnniRouteNode OptionalGroup	pnniRouteNodeDestPortId, pnniRouteNodeProto, pnniRouteNodeTimeStamp, pnniRouteNodeInfo, pnniRouteNodeGcacClp, pnniRouteNodeFwdMetricAW, pnniRouteNodeFwdMetric1, pnniRouteNodeFwdMetric2, pnniRouteNodeFwdMetric3, pnniRouteNodeFwdMetric4, pnniRouteNodeFwdMetric5, pnniRouteNodeFwdMetric6, pnniRouteNodeFwdMetric7, pnniRouteNodeFwdMetric8, pnniRouteNodeBwdMetricAW, pnniRouteNodeBwdMetric1, pnniRouteNodeBwdMetric2, pnniRouteNodeBwdMetric3, pnniRouteNodeBwdMetric4, pnniRouteNodeBwdMetric5, pnniRouteNodeBwdMetric6, pnniRouteNodeBwdMetric7, pnniRouteNodeBwdMetric8, pnniRouteNodeVPCapability, pnniRouteNodeStatus	Specifies a collection of the optional PNNI objects used to manage precalculated routes to nodes in the PNNI routing domain. Status: current	none
pnniDTLOptional Group	pnniDTLNodeId, pnniDTLPortId, pnniDTLLinkType, pnniDTLStatus	Specifies a collection of the optional PNNI objects used to manage the precalculated routes to nodes in the PNNI routing domain. Status: current	none

Table 5-27 ATM PNNI MIB Objects Used for Units of Conformance (continued)

Name	Objects	Description	Default Value
pnniRouteAddr OptionalGroup	pnniRouteAddrIfIndex, pnniRouteAddrAdvertisingNodeId, pnniRouteAddrAdvertisedPortId, pnniRouteAddrType, pnniRouteAddrProto, pnniRouteAddrPnniScope, pnniRouteAddrVPCapability, pnniRouteAddrMetricsTag, pnniRouteAddrPtseId, pnniRouteAddrOriginate Advertisement, pnniRouteAddrInfo, pnniRouteAddrOperStatus, pnniRouteAddrTimeStamp, pnniRouteAddrRowStatus	Specifies a collection of the optional PNNI objects used to manage the routes to the reachable addresses in the PNNI routing domain. Status: current	none
pnniRouteTnsOptional Group	pnniRouteTnsIfIndex, pnniRouteTnsAdvertisingNodeId, pnniRouteTnsAdvertisedPortId, pnniRouteTnsRouteType, pnniRouteTnsProto, pnniRouteTnsPnniScope, pnniRouteTnsVPCapability, pnniRouteTnsMetricsTag, pnniRouteTnsPtseId, pnniRouteTnsOriginateAdvertisement, pnniRouteTnsInfo, pnniRouteTnsOperStatus, pnniRouteTnsTimeStamp, pnniRouteTnsRowStatus	Specifies a collection of the optional PNNI objects used to manage the routes to the reachable transit networks in the PNNI routing domain. Status: current	none

Cisco WAN SVC MIB Objects

The Cisco WAN SVC MIB objects configure, provision, and gather status for ATM services. This MIB file is defined in the CISCO-WAN-SVC-MIB.my file.

Note The Cisco WAN SVC MIB is defined under the StrataCom Enterprise.

Enhanced Network to Network Interface (ENNI) indicates an enhanced Network-to-Network Interface (NNI) to link two networks using either Cisco Automatic Routing Management (AR) or PNNI. Switched or Permanent Virtual Circuit (XPVC), end-to-end Virtual Circuits (VCs), are provisioned across multiple networks to support AR and PNNI. XPVC consists of multiple segments where each segment is a complete VC, which can be a PVC or SPVC. Within the Cisco WAN SVC MIB objects, XPVC changes the segment Operation and Maintenance (OAM) cell to handle the flow for end-to-end OAM segment loops. For example, two or three segments can be between the user ports. When ENNI is defined as an LMI function, auto-discovery is enabled for ENNI.

The Cisco WAN SVC MIB objects include:

- Service Information Group
- Interface Configuration Table
- Port Call Statistics Table

- Port Address Table
- Port Connection Admission Control Configuration Table
- Port Loading Table
- Interface Operation Table
- Port Signaling Statistics
- Port Connection Data Table
- Port Connection Trace Control Table
- Cisco WAN SVC Conformance and Compliance Statements

Service Information Group

The object identifier for each MIB object is listed in Table 5-28.

Table 5-28 Service Information Group Object Identifiers

Name	Object Identifier
cwsSwRevision	::= { ciscoWANSvcInfo 1 }
cwsControllerStatus	::= { ciscoWANSvcInfo 6 }
cwspPnniStndbyControllerStatus	::= { ciscoWANSvcInfo 7 }
cwspPnniControllerStatus	::= { ciscoWANSvcInfo 8 }
cwspPnniControllerPhySlot	::= { ciscoWANSvcInfo 9 }

The MIB objects are listed in Table 5-29.

Table 5-29 Service Information Group MIB Objects

Name	Syntax	Description	Default Value
cwsSwRevision	`DisplayString`	Determines the PNNI network controller software revision number. Max-Access: read-only Status: current	none
cwsControllerStatus	`CwspControllerState`	Specifies the administrative status for the PNNI active controller. Max-Access: read-only Status: current	none
cwspPnniStndbyController Status	`CwspControllerState`	Specifies the administrative status for the standby controller. Max-Access: read-only Status: current	none

Table 5-29 Service Information Group MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspPnniControllerStatus	`Integer {active (1), stndby (2), unavailable(3)}`	Specifies the administrative status of the PNNI controller. **Note** This object is used only in the trap varbind; the SNMP `Get` operation is meaningless. Max-Access: read-only Status: current	none
cwspPnniControllerPhySlot	`Integer32`	Specifies the PNNI controller physical location. **Note** This object is used only in the trap varbind; the SNMP `Get` operation is meaningless. Max-Access: read-only Status: current	none

Interface Configuration Table

The object identifier for each MIB object is listed in Table 5-30.

Table 5-30 Interface Configuration Table Object Identifiers

Name	Object Identifier
cwspConfigTable	::= { cwspConfig 1 }
cwspConfigEntry	::= { cwspConfigTable 1 }
cwspAdminStatus	::= { cwspConfigEntry 1 }
cwspOperStatus	::= { cwspConfigEntry 2 }
cwspSvcBlocked	::= { cwspConfigEntry 3 }
cwspSpvcBlocked	::= { cwspConfigEntry 4 }
cwspIlmiAddrRegEnable	::= { cwspConfigEntry 5 }
cwspIlmiAutoConfEnable	::= { cwspConfigEntry 6 }
cwspIlmiServRegEnable	::= { cwspConfigEntry 7 }
cwspPhyIdentifier	::= { cwspConfigEntry 8 }
cwspSignallingVpi	::= { cwspConfigEntry 9 }
cwspSignallingVci	::= { cwspConfigEntry 10 }
cwspRoutingVpi	::= { cwspConfigEntry 11 }
cwspRoutingVci	::= { cwspConfigEntry 12 }
cwspMaxVpiBits	::= { cwspConfigEntry 13 }
cwspMaxVciBits	::= { cwspConfigEntry 14 }
cwspUniVersion	::= { cwspConfigEntry 15 }
cwspNniVersion	::= { cwspConfigEntry 16 }

Table 5-30 Interface Configuration Table Object Identifiers (continued)

Name	Object Identifier
cwspUniType	::= { cwspConfigEntry 17 }
cwspSide	::= { cwspConfigEntry 18 }
cwspMaxP2pCalls	::= { cwspConfigEntry 19 }
cwspMaxP2mpRoots	::= { cwspConfigEntry 20 }
cwspMaxP2mpLeafs	::= { cwspConfigEntry 21 }
cwspMinSvccVpi	::= { cwspConfigEntry 22 }
cwspMaxSvccVpi	::= { cwspConfigEntry 23 }
cwspMinSvccVci	::= { cwspConfigEntry 24 }
cwspMaxSvccVci	::= { cwspConfigEntry 25 }
cwspMinSvpcVpi	::= { cwspConfigEntry 26 }
cwspMaxSvpcVpi	::= { cwspConfigEntry 27 }
cwspEnhancedIisp	::= { cwspConfigEntry 28 }
cwspConfigTableRowStatus	::= { cwspConfigEntry 29 }
cwspAddrPlanSupported	::= { cwspConfigEntry 30 }
cwspIlmiSecureLink	::= { cwspConfigEntry 31 }
cwspIlmiAttachmentPoint	::= { cwspConfigEntry 32 }
cwspIlmiLocalAttrStd	::= { cwspConfigEntry 33 }
cwspIlmiUCSMEnable	::= { cwspConfigEntry 34 }

The MIB objects are listed in Table 5-31.

Table 5-31 Interface Configuration Table MIB Objects

Name	Syntax	Description	Default Value
cwspConfigTable	`Sequence of CwspConfigEntry`	Collects attributes that affect the operation of the controller interface. There is a single row for each interface that the managed system expects to add or manage. Max-Access: not-accessible Status: current	none
cwspConfigEntry	`CwspConfigEntry`	Specifies an entry in the interface configuration table that contains information about an interface. Max-Access: not-accessible Status: current The index contains `ifIndex`.	none

Table 5-31 Interface Configuration Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspAdminStatus	`INTEGER {inService (1), outService (2)}`	Specifies the administrative status of the interface. `inService (1)` means that the interface is operational, whereas, `outService (2)` indicates that the interface is put out of service by the administrator. Max-Access: read-create Status: current	`outservice`
cwspOperStatus	`INTEGER {ok(1), failed(2), other(3)}`	Specifies the operational state of the interface. This status is derived from the physical state of the interface. Values are defined as follows: • `failed (1)`—Indicates that the interface has failed. • `ok (2)`—Indicates that the interface is in an operational state. • `other (3)`—Indicates that the interface is in an unknown state. Max-Access: read-only Status: current	none
cwspSvcBlocked	`TruthValue`	Indicates whether virtual connections are allowed through this interface. Max-Access: read-create Status: current	`false`
cwspSpvcBlocked	`TruthValue`	Indicates whether SPVCs are allowed through this interface. Max-Access: read-create Status: current	`false`
cwspIlmiAddrReg Enable	`TruthValue`	Indicates whether the Interim Local Management Interface (ILMI) address registration is enabled or disabled. Max-Access: read-create Status: current	`true`
cwspIlmiAutoConf Enable	`TruthValue`	Indicates whether the auto configuration of the interface is turned on or off. If auto configuration is enabled, the interface comes up using the ILMI auto-configuration. Max-Access: read-create Status: current	`true`

Table 5-31 Interface Configuration Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspIlmiServRegEnable	TruthValue	Indicates whether the service registry is enabled or disabled on the interface. Max-Access: read-create Status: current	true
cwspPhyIdentifier	DisplayString	Indicates the physical identification of the interface. A physical identification is mandatory when the port is provisioned for the first-time through SNMP. Max-Access: read-create Status: current	none
cwspSignallingVpi	Integer32 (0..4095)	Denotes the signaling Virtual Path Identifier (VPI) that is used on the interface. Max-Access: read-create Status: current	0
cwspSignallingVci	Integer32 (0..65535)	Indicates the signaling Virtual Channel Identifier (VCI) that is used on the interface. The acceptable values are from 32 to 65535. Max-Access: read-create Status: current	5
cwspRoutingVpi	Integer32 (0..4095)	Indicates the VPI that is used for the PNNI lowest-level RCC. Max-Access: read-create Status: current	0
cwspRoutingVci	Integer32 (0..65535)	Indicates the VCI that is used for the PNNI lowest-level RCC. The acceptable values are 32 to 65535. Max-Access: read-create Status: current	18
cwspMaxVpiBits	Integer32 (0..12)	Determines the maximum number of active VPI bits on this ATM Interface. For virtual interfaces, for example, Virtual Path Connections used by PNNI, this value has no meaning and is set to 0. Max-Access: read-only Status: current	none
cwspMaxVciBits	Integer32 (0..16)	Determines the maximum number of active VCI bits on this ATM Interface. Max-Access: read-only Status: current	none

Table 5-31 Interface Configuration Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspUniVersion	`Integer {uni20 (1), uni30 (2), uni31 (3), uni40 (4), ituDss2 (5), frf4 (6), unsupported (7), ip (8), self(9)}`	Indicates the latest version of the *ATM Forum UNI Signaling Specification* that is supported on this ATM Interface. If this value is not present, a version of the UNI earlier than 3.1 is supported. The value of `self(9)` indicates the entry is a voice port. If the peer Interface Management Entity (IME) value of this object is the same as, or later than, the local IME value, the version corresponding to the local IME value can be attempted. Otherwise, if the peer IME value of this object is earlier and supported locally, the local IME can use the version that corresponds to the peer IME value. Otherwise, compatibility of the two IMEs cannot be assumed. When this variable has a value other than `unsupported(7)`, the `cswpNniVersion` variable contains the value `unsupported(7)`. Max-Access: read-create Status: current	`uni31`
cwspNniVersion	`Integer{iisp30(1), iisp31(2), pnni10(3), enni(4), aini(5) unsupported(7)}`	Indicates the latest version of the *ATM Forum PNNI Signaling Specification* that is supported on this ATM Interface. **Note** The PNNI routing version is not determined through ILMI. If the peer IME value of this object is the same as, or later than, the local IME value, the version that corresponds to the local IME value is attempted. Otherwise, if the peer IME value of this object is earlier, and supported locally, the local IME can use the version that corresponds to the peer IME value. Otherwise, compatibility of the two IMEs cannot be assumed. When this variable has a value other than `unsupported(7)`, the `cswpUniVersion` variable contains the value `unsupported(7)`. **Note** `enni` is not a signaling protocol. Max-Access: read-create Status: current	`pnni10`

Table 5-31 Interface Configuration Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspUniType	`INTEGER { public(1), private(2) }`	Specifies the type of ATM device, for example, public or private. Max-Access: read-create Status: current	`private`
cwspSide	`INTEGER { user(1), network(2) }`	Determines the type of ATM device. This object is used in an automatic ATM Interface-Type determination procedure, which determines a correct operational ATM Interface-type. An ATM end system takes the value `ofuser(1)`; an ATM network node takes the value of `node(2)`. Max-Access: read-create Status: current	`network`
cwspMaxP2pCalls	`Integer32 (0..65535)`	Determines the maximum number of point-to-point calls. The calls include Virtual Circuit (VC) and Virtual Path (VP) allowed on the interface. Max-Access: read-create Status: current	`10000`
cwspMaxP2mpRoots	`Integer32 (0..65535)`	Indicates the maximum number of root VCs, for example, point-to-multipoint, is allowed on the interface. Max-Access: read-create Status: current	`1000`
cwspMaxP2mpLeafs	`Integer32 (0..65535)`	Indicates the maximum number of leaf VCs, for example, point-to-multipoint, allowed on the interface. Max-Access: read-create Status: current	`4095`
cwspMinSvccVpi	`Integer32 (0..4095)`	Indicates the minimum SVCC Virtual Path Identifier (VPI) configured on the interface. Max-Access: read-create Status: current	`0`
cwspMaxSvccVpi	`Integer32 (0..4095)`	Indicates the maximum SVCC VPI configured on the interface. Max-Access: read-create Status: current	`4095`
cwspMinSvccVci	`Integer32 (0..65535)`	Indicates the minimum SVCC Virtual Channel Identifier (VCI) configured on the interface. Max-Access: read-create Status: current	`35`

Table 5-31 Interface Configuration Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspMaxSvccVci	`Integer32 (35..65535)`	Indicates the maximum SVCC VCI configured on the interface. Max-Access: read-create Status: current	`65535`
cwspMinSvpcVpi	`Integer32 (1..4095)`	Indicates the minimum SVPC VPI configured on the interface. Max-Access: read-create Status: current	`1`
cwspMaxSvpcVpi	`Integer32 (1..4095)`	Indicates the maximum SVPC VPI configured on the interface. Max-Access: read-create Status: current	`4095`
cwspEnhancedIisp	`TruthValue`	Indicates whether the enhanced features for Interim-Interswitch Signaling Protocol (IISP) are enabled or disabled. Max-Access: read-create Status: current	`false`
cwspConfigTableRow Status	`RowStatus`	Creates and deletes the interface. The values `createAndGo`, `destroy`, `active`, and `notInService` are supported. The NMS uses only `createAndGo` to create a row and `destroy` to delete a row. The managed device can either return `active` or `notInService` for a row status. Max-Access: read-create Status: current	`none`
cwspAddrPlanSupported	`Integer{ both(1), aesa(2), e164(3) }`	Specifies the ATM address plan that is supported on an interface. This is modified only if the interface is a public UNI. For all other interfaces, the value is `aesa`. Max-Access: read-create Status: current	`aesa`
cwspIlmiSecureLink	`TruthValue`	Indicates whether the ILMI Secure Link Protocol is enabled or disabled. When the secure link protocol is enabled, loss in the ILMI connectivity is treated as loss in the attachment point. Therefore, all the SVCs and SVPs are released on the interface. Max-Access: read-create Status: current	`true`

Table 5-31 Interface Configuration Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspIlmiAttachment Point	TruthValue	Indicates whether detection of the loss of attachment procedures is enabled on this interface. When set to true, the standard ILMI procedures are employed to detect the loss of the attachment point. If set to false, the ILMI protocol on the interface does not detect the loss of the attachment. Max-Access: read-create Status: current	true
cwspIlmiLocalAttrStd	TruthValue	Indicates whether modification of the local attributes is used for either the ILMI 4.0 specification or Cisco proprietary procedures. When set to true, the standard ILMI procedures are followed. Max-Access: read-create Status: current	true
cwspIlmiUCSMEnable	TruthValue	Indicates whether the ILMI user-connection status monitor is enabled or disabled. Max-Access: read-create Status: current	true

Port Call Statistics Table

The object identifier for each MIB object is listed in Table 5-32.

Table 5-32 Port Call Statistics Table Object Identifiers

Name	Object Identifier
cwspCallStatsTable	::= { cwspCallStats 1 }
cwspCallStatsEntry	::= { cwspCallStatsTable 1 }
cwspCountReset	::= { cwspCallStatsEntry 1 }
cwspInCallAttempts	::= { cwspCallStatsEntry 2 }
cwspInCallEstabs	::= { cwspCallStatsEntry 3 }
cwspInCallFailures	::= { cwspCallStatsEntry 4 }
cwspInFilterFailures	::= { cwspCallStatsEntry 5 }
cwspInRouteFailures	::= { cwspCallStatsEntry 6 }
cwspInResrcFailures	::= { cwspCallStatsEntry 7 }
cwspInTimerFailures	::= { cwspCallStatsEntry 8 }
cwspInCrankbacks	::= { cwspCallStatsEntry 9 }
cwspOutCallAttempts	::= { cwspCallStatsEntry 10 }

Table 5-32 Port Call Statistics Table Object Identifiers (continued)

Name	Object Identifier
cwspOutCallEstabs	::= { cwspCallStatsEntry 11 }
cwspOutCallFailures	::= { cwspCallStatsEntry 12 }
cwspOutFilterFailures	::= { cwspCallStatsEntry 13 }
cwspOutRouteFailures	::= { cwspCallStatsEntry 14 }
cwspOutResrcFailures	::= { cwspCallStatsEntry 15 }
cwspOutTimerFailures	::= { cwspCallStatsEntry 16 }
cwspOutCrankbacks	::= { cwspCallStatsEntry 17 }

The MIB objects are listed in Table 5-33.

Table 5-33 Port Call Statistics Table MIB Objects

Name	Syntax	Description	Default Value
cwspCallStatsTable	`Sequence of CwspCallStatsEntry`	Contains objects that shows the statistics for SVC and SPVC calls on a given interface. Max-Access: not-accessible Status: current	none
cwspCallStatsEntry	`CwspCallStatsEntry`	Contains an entry that is used for SVC and SPVC call statistics. Max-Access: not-accessible Status: current The index contains `ifIndex`.	none
cwspCountReset	`INTEGER {noop(1), -- none of following reset(2) -- resetting}`	Writes a value of two to this object to reset all the counters. Max-Access: read-write Status: current	`noop`
cwspInCallAttempts	`Counter32`	Displays the number of incoming signaling messages `Setup` and `Add Party` that are received by the switching node on this interface. These messages are used for call establishment. Max-Access: read-only Status: current	none
cwspInCallEstabs	`Counter32`	Displays the number of incoming signaling messages `Connect` and `Add Party Ack` that are received by the switching node on this interface. These messages mark a call that is successfully established. Max-Access: read-only Status: current	none

Table 5-33 Port Call Statistics Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspInCallFailures	Counter32	Displays the total number of incoming point-to-point and point-to-multipoint SVC/SPVC call attempts that failed on this interface. Max-Access: read-only Status: current	none
cwspInFilterFailures	Counter32	Displays the number of incoming point-to-point and point-to-multipoint SVC/SPVC call attempts that failed due to address filtering on this interface. Max-Access: read-only Status: current	none
cwspInRouteFailures	Counter32	Displays the number of incoming point-to-point and point-to-multipoint SVC/SPVC call attempts on this interface. These attempts failed because a route to the destination was not available. Max-Access: read-only Status: current	none
cwspInResrcFailures	Counter32	Displays the number of incoming point-to-point and point-to-multipoint SVC/SPVC call attempts on this interface. These attempts failed because there were not enough resources as requested in call parameters. Max-Access: read-only Status: current	none
cwspInTimerFailures	Counter32	Displays the number of signaling timers timed out for incoming point-to-point and point-to-multipoint SVC/SPVC calls on this interface. Max-Access: read-only Status: current	none
cwspInCrankbacks	Counter32	Displays the number of crankback Information Elements (IEs) received on this interface for incoming point-to-point and point-to-multipoint SVC/SPVC call attempts. Max-Access: read-only Status: current	none
cwspOutCallAttempts	Counter32	Displays the number of outgoing signaling messages `Setup` and `Add Party` on this interface. These messages are used for call establishment. Max-Access: read-only Status: current	none

Table 5-33 Port Call Statistics Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspOutCallEstabs	Counter32	Displays the number of outgoing signaling messages Connect and Add Party Ack, which mark the call being established on this interface. Max-Access: read-only Status: current	none
cwspOutCallFailures	Counter32	Displays the total number of outgoing signaling messages for both point-to-point and point-to-multipoint call establishments on this interface that failed. Max-Access: read-only Status: current	none
cwspOutFilterFailures	Counter32	Displays the number of outgoing signaling messages that are used for call establishment on this interface, and which failed the address filtering. Max-Access: read-only Status: current	none
cwspOutRouteFailures	Counter32	Displays the number of outgoing signaling messages that are used for call establishment on this interface. These messages failed because the route was not available. Max-Access: read-only Status: current	none
cwspOutResrcFailures	Counter32	Displays the number of outgoing signaling messages that are used for call establishment on this interface. These messages failed because the resource request was not available. Max-Access: read-only Status: current	none
cwspOutTimerFailures	Counter32	Displays the number of timed-out signaling timers on this interface that are used for outgoing signaling messages. Max-Access: read-only Status: current	none
cwspOutCrankbacks	Counter32	Displays the number of crankback IEs being sent on this interface that are used for outgoing signaling release messages. This is generated on the node that generates the crankback IEs. Max-Access: read-only Status: current	none

Port Address Table

The object identifier for each MIB object is listed in Table 5-34.

Table 5-34 Port Address Table Object Identifiers

Name	Object Identifier
cwspAddressTable	::= { cwspAddress 1 }
cwspAddressEntry	::= { cwspAddressTable 1 }
cwspAtmAddress	::= { cwspAddressEntry 1 }
cwspAddrLen	::= { cwspAddressEntry 2 }
cwspAddrType	::= { cwspAddressEntry 3 }
cwspAddrProto	::= { cwspAddressEntry 4 }
cwspAddrPlan	::= { cwspAddressEntry 5 }
cwspAddrScope	::= { cwspAddressEntry 6 }
cwspAddrRedistribute	::= { cwspAddressEntry 7 }
cwspAddressRowStatus	::= { cwspAddressEntry 8 }

The MIB objects are listed in Table 5-35.

Table 5-35 Port Address Table MIB Objects

Name	Syntax	Description	Default Value
cwspAddressTable	`Sequence of CwspAddressEntry`	Contains all the attributes necessary to determine what the PNNI entity recognizes are reachable in terms of the ATM End System Addresses, and which nodes are advertising this ability. This interface ATM Address Table is also used to configure static routes to reachable addresses. Entries in this table can be created or deleted by setting the `intfAddressRowStatus` object to appropriate values. Existing entries in this table cannot be modified. Entries in this table can also be created or deleted through the command provided by the Command Line Interface (CLI). Max-Access: not-accessible Status: current	none
cwspAddressEntry	`CwspAddressEntry`	Specifies an entry in the table that contains information about a reachable address. Max-Access: not-accessible Status: current The following list the indexes: • `ifIndex` • `cwspAtmAddress` • `cwspAddrLen`	none
cwspAtmAddress	`AtmAddress`	Determines the value of the ATM End System Address. Max-Access: not-accessible Status: current	none
cwspAddrLen	`INTEGER (0..160)`	Determines the address length in bits to be applied to the ATM End System Address. Max-Access: not-accessible Status: current	none

Table 5-35 Port Address Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspAddrType	`INTEGER { internal(1), exterior(2) }`	Specifies the type of reachability from the advertising node to the address. This variable indicates that either an internal ATM address is a reachable ATM destination that is directly attached to the logical node, which is advertising this address, or that an exterior ATM address is reached through a PNNI routing domain, which is not located in this routing domain. Refer to *ATM Forum PNNI 1.0*, Section 5.8.1.3. Max-Access: read-create Status: current	`exterior`
cwspAddrProto	`INTEGER { local(1), static(2) }`	Specifies the routing mechanism through the connectivity from the advertising node to the learned reachable address. The following is how each address is defined: • `local`—Indicates this address is automatically advertised. • `static`—Indicates this address is not automatically advertised. Setting the `cwspAddrRedistribute` variable to true forces a static address to be advertised. Max-Access: read-create Status: current	`local`
cwspAddrPlan	`INTEGER { e164(1), nsap(2) }`	Determines the address plan. For the Network Service Access Point (NSAP) address, the first byte of the address automatically implies one of the three NSAP address plans: NSAP E.164, NSAP DCC, or NSAP ICD. Max-Access: read-create Status: current	`nsap`
cwspAddrScope	`INTEGER (0..104)`	Specifies the PNNI scope of advertisement, for example, a level of the PNNI hierarchy which is used for the reachability from the advertising node to the address. Max-Access: read-create Status: current	`0`

Table 5-35 Port Address Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspAddrRedistribute	`TruthValue`	Defines whether or not the reachable address specified by this entry is advertised by the local node into the PNNI routing domain. This object is only meaningful if the routing mechanism `cwspAddrProto` is `static`. Max-Access: read-create Status: current	`false`
cwspAddressRowStatus	`RowStatus`	Creates or deletes a reachable address. **Note** NMS can use `CreateAndGo` to create a row and `Destroy` to delete a row. The managed device returns an `active` or `notInService` as row status. Max-Access: read-create Status: current	none

Port Connection Admission Control Configuration Table

The object identifier for each MIB object that is used for the Port Connection Admission Control (CAC) Configuration Table is listed in Table 5-36.

Table 5-36 Port CAC Configuration Table Object Identifiers

Name	Object Identifier
cwspCacConfigTable	::= { cwspCacConfig 1 }
cwspCacConfigEntry	::= { cwspCacConfigTable 1 }
cwspUtilFactorCbr	::= { cwspCacConfigEntry 1 }
cwspUtilFactorRtVbr	::= { cwspCacConfigEntry 2 }
cwspUtilFactorNrtVbr	::= { cwspCacConfigEntry 3 }
cwspUtilFactorAbr	::= { cwspCacConfigEntry 4 }
cwspUtilFactorUbr	::= { cwspCacConfigEntry 5 }
cwspMaxBwCbr	::= { cwspCacConfigEntry 6 }
cwspMaxBwRtVbr	::= { cwspCacConfigEntry 7 }
cwspMaxBwNrtVbr	::= { cwspCacConfigEntry 8 }
cwspMaxBwAbr	::= { cwspCacConfigEntry 9 }
cwspMaxBwUbr	::= { cwspCacConfigEntry 10 }
cwspMinBwCbr	::= { cwspCacConfigEntry 11 }
cwspMinBwRtVbr	::= { cwspCacConfigEntry 12 }
cwspMinBwNrtVbr	::= { cwspCacConfigEntry 13 }

Table 5-36 Port CAC Configuration Table Object Identifiers (continued)

Name	Object Identifier
cwspMinBwAbr	::= { cwspCacConfigEntry 14 }
cwspMinBwUbr	::= { cwspCacConfigEntry 15 }
cwspMaxVcCbr	::= { cwspCacConfigEntry 16 }
cwspMaxVcRtVbr	::= { cwspCacConfigEntry 17 }
cwspMaxVcNrtVbr	::= { cwspCacConfigEntry 18 }
cwspMaxVcAbr	::= { cwspCacConfigEntry 19 }
cwspMaxVcUbr	::= { cwspCacConfigEntry 20 }
cwspMinVcCbr	::= { cwspCacConfigEntry 21 }
cwspMinVcRtVbr	::= { cwspCacConfigEntry 22 }
cwspMinVcNrtVbr	::= { cwspCacConfigEntry 23 }
cwspMinVcAbr	::= { cwspCacConfigEntry 24 }
cwspMinVcUbr	::= { cwspCacConfigEntry 25 }
cwspMaxVcBwCbr	::= { cwspCacConfigEntry 26 }
cwspMaxVcBwRtVbr	::= { cwspCacConfigEntry 27 }
cwspMaxVcBwNrtVbr	::= { cwspCacConfigEntry 28 }
cwspMaxVcBwAbr	::= { cwspCacConfigEntry 29 }
cwspMaxVcBwUbr	::= { cwspCacConfigEntry 30 }
cwspDefaultCdvtCbr	::= { cwspCacConfigEntry 31 }
cwspDefaultCdvtRtVbr	::= { cwspCacConfigEntry 32 }
cwspDefaultCdvtNrtVbr	::= { cwspCacConfigEntry 33 }
cwspDefaultCdvtAbr	::= { cwspCacConfigEntry 34 }
cwspDefaultCdvtUbr	::= { cwspCacConfigEntry 35 }
cwspDefaultMbsRtVbr	::= { cwspCacConfigEntry 36 }
cwspDefaultMbsNrtVbr	::= { cwspCacConfigEntry 37 }

The MIB objects are listed in Table 5-37.

Table 5-37 Port CAC Configuration Table MIB Objects

Name	Syntax	Description	Default Value
cwspCacConfigTable	`Sequence of CwspCacConfigEntry`	Specifies a table that contains the CAC information for each interface. Max-Access: not-accessible Status: current	none
cwspCacConfigEntry	`CwspCacConfigEntry`	Contains an entry for the CAC information. Max-Access: not-accessible Status: current The index contains `ifIndex`.	none
cwspUtilFactorCbr	`Integer32 (1..200)`	Specifies the booking factor for the Constant Bit Rate (CBR) service. Max-Access: read-write Status: current	`100`
cwspUtilFactorRtVbr	`Integer32 (1..200)`	Specifies the booking factor for the real-time Variable Bit Rate (VBR) service. Max-Access: read-write Status: current	`100`
cwspUtilFactorNrtVbr	`Integer32 (1..200)`	Specifies the booking factor for nonreal-time VBR service. Max-Access: read-write Status: current	`100`
cwspUtilFactorAbr	`Integer32 (1..200)`	Specifies the booking factor for the Available Bit Rate (ABR) service. Max-Access: read-write Status: current	`100`
cwspUtilFactorUbr	`Integer32 (1..200)`	Specifies the booking factor for the Unspecified Bit Rate (UBR) service. Max-Access: read-write Status: current	`100`
cwspMaxBwCbr	`Integer32 (0..1000000)`	Specifies the maximum bandwidth for the CBR service in percentage. The value of this variable is interpreted in the format of *xxx.xxxx*. For example, *xxx.xxxx* is a value 750000 and is interpreted as 75.0000%. Max-Access: read-write Status: current	`1000000`

Table 5-37 Port CAC Configuration Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspMaxBwRtVbr	`Integer32 (0..1000000)`	Specifies the maximum bandwidth for the real-time VBR service in percentage. The value of this variable is interpreted in the format of *xxx.xxx*. For example, *xxx.xxxx* is a value 750000 and is interpreted as 75.0000%. Max-Access: read-write Status: current	`1000000`
cwspMaxBwNrtVbr	`Integer32 (0..1000000)`	Specifies the maximum bandwidth for the nonreal-time (NRT) VBR service in percentage. The value of this variable is interpreted in the format of *xxx.xxx*. For example, *xxx.xxxx* is a value 750000 and is interpreted as 75.0000%. Max-Access: read-write Status: current	`1000000`
cwspMaxBwAbr	`Integer32 (0..1000000)`	Specifies the maximum bandwidth for the ABR service in percentage. The value of this variable is interpreted in the format of *xxx.xxx*. For example, *xxx.xxxx* is a value 750000 and is interpreted as 75.0000%. Max-Access: read-write Status: current	`1000000`
cwspMaxBwUbr	`Integer32 (0..1000000)`	Specifies the maximum bandwidth for the UBR service in percentage. The value of this variable is interpreted in the format of *xxx.xxx*. For example, *xxx.xxxx* is a value 750000 and is interpreted as 75.0000%. Max-Access: read-write Status: current	`1000000`
cwspMinBwCbr	`Integer32 (0..1000000)`	Specifies the minimum bandwidth for the CBR service in percentage. The total values for `cwspMinBwCbr`, `cwspMinBwRtVbr`, `cwspMinBwNrtVbr`, `cwspMinBwAbr`, and `cwspMinBwUbr` cannot exceed 1000000 or 100%. The value of this variable is interpreted in the format of *xxx.xxx*. For example, *xxx.xxxx* is a value 750000 and is interpreted as 75.0000%. Max-Access: read-write Status: current	`0`

Table 5-37 Port CAC Configuration Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspMinBwRtVbr	`Integer32 (0..1000000)`	Specifies the minimum bandwidth for the real-time VBR service in percentage. The total values of `cwspMinBwCbr`, `cwspMinBwRtVbr`, `cwspMinBwNrtVbr`, `cwspMinBwAbr`, and `cwspMinBwUbr` can not exceed 1000000 or 100%. The value of this variable is interpreted in the format of *xxx.xxx*. For example, *xxx.xxxx* is a value 750000 and is interpreted as 75.0000%. Max-Access: read-write Status: current	`0`
cwspMinBwNrtVbr	`Integer32 (0..1000000)`	Specifies the minimum bandwidth for the NRT VBR service in percentage. The total values of `cwspMinBwCbr`, `cwspMinBwRtVbr`, `cwspMinBwNrtVbr`, `cwspMinBwAbr`, and `cwspMinBwUbr` cannot exceed 1000000 or 100%. The value of this variable is interpreted in the format of *xxx.xxx*. For example, *xxx.xxxx* is a value 750000 and is interpreted as 75.0000%. Max-Access: read-write Status: current	`0`
cwspMinBwAbr	`Integer32 (0..1000000)`	Specifies the minimum bandwidth for the ABR service in percentage. The total values of `cwspMinBwCbr`, `cwspMinBwRtVbr`, `cwspMinBwNrtVbr`, `cwspMinBwAbr`, and `cwspMinBwUbr` cannot exceed 1000000 or 100%. The value of this variable is interpreted in the format of *xxx.xxx*. For example, *xxx.xxxx* is a value 750000 and is interpreted as 75.0000%. Max-Access: read-write Status: current	`0`
cwspMinBwUbr	`Integer32(0..1000000)`	Specifies the minimum bandwidth for the UBR service in percentage. The value for this variable is always 0. Max-Access: read-only Status: current	none

Table 5-37 Port CAC Configuration Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspMaxVcCbr	`Integer32 (0..1000000)`	Specifies the maximum number of VCs for the CBR service in percentage. The value of this variable is interpreted in the format of *xxx.xxx*. For example, *xxx.xxxx* is a value 750000 and is interpreted as 75.0000%. Max-Access: read-write Status: current	`1000000`
cwspMaxVcRtVbr	`Integer32 (0..1000000)`	Specifies the maximum number of VCs for the real-time VBR service in percentage. The value of this variable is interpreted in the format of *xxx.xxx*. For example, *xxx.xxxx* is a value 750000 and is interpreted as 75.0000%. Max-Access: read-write Status: current	`1000000`
cwspMaxVcNrtVbr	`Integer32 (0..1000000)`	Specifies the maximum number of VCs for the NRT VBR service in percentage. The value of this variable is interpreted in the format of *xxx.xxx*. For example, *xxx.xxxx* is a value 750000 and is interpreted as 75.0000%. Max-Access: read-write Status: current	`1000000`
cwspMaxVcAbr	`Integer32 (0..1000000)`	Specifies the maximum number of VCs for the ABR service in percentage. The value of this variable is interpreted in the format of *xxx.xxx*. For example, *xxx.xxxx* is a value 750000 and is interpreted as 75.0000%. Max-Access: read-write Status: current	`1000000`

Table 5-37 Port CAC Configuration Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspMaxVcUbr	`Integer32 (0..1000000)`	Specifies the maximum number of VCs for the UBR service in percentage. The value of this variable is interpreted in the format of *xxx.xxx*. For example, *xxx.xxxx* is a value 750000 and is interpreted as 75.0000%. Max-Access: read-write Status: current	`1000000`
cwspMinVcCbr	`Integer32 (0..1000000)`	Specifies the minimum number of VCs for the CBR service in percentage. The total values of `cwspMinVcCbr`, `cwspMinVcRtVbr`, `cwspMinVcNrtVbr`, `cwspMinVcAbr`, and `cwspMinVcUbr` cannot exceed 1000000 or 100%. The value of this variable is interpreted in the format of *xxx.xxx*. For example, *xxx.xxxx* is a value 750000 and is interpreted as 75.0000%. Max-Access: read-write Status: current	`0`
cwspMinVcRtVbr	`Integer32 (0..1000000)`	Specifies the minimum number of VCs for the real-time VBR service in percentage. The total values of `cwspMinVcCbr`, `cwspMinVcRtVbr`, `cwspMinVcNrtVbr`, `cwspMinVcAbr`, and `cwspMinVcUbr` cannot exceed 1000000 or 100%. The value of this variable is interpreted in the format of *xxx.xxx*. For example, *xxx.xxxx* is a value 750000 and is interpreted as 75.0000%. Max-Access: read-write Status: current	`0`

Table 5-37 Port CAC Configuration Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspMinVcNrtVbr	`Integer32 (0..1000000)`	Specifies the minimum number of VCs for the NRT VBR service in percentage. The total values of `cwspMinVcCbr`, `cwspMinVcRtVbr`, `cwspMinVcNrtVbr`, `cwspMinVcAbr`, and `cwspMinVcUbr` cannot exceed 1000000, for example, 100%. The value of this variable is interpreted in the format of *xxx.xxx*. For example, *xxx.xxxx* is a value 750000 and is interpreted as 75.0000%. Max-Access: read-write Status: current	`0`
cwspMinVcAbr	`Integer32 (0..1000000)`	Specifies the minimum number of VCs for the ABR service in percentage. The total values of `cwspMinVcCbr`, `cwspMinVcRtVbr`, `cwspMinVcNrtVbr`, `cwspMinVcAbr`, and `cwspMinVcUbr` cannot exceed 1000000, for example, 100%. The value of this variable is interpreted in the format of *xxx.xxx*. For example, *xxx.xxxx* is a value 750000 and is interpreted as 75.0000%. Max-Access: read-write Status: current	`0`
cwspMinVcUbr	`Integer32 (0..1000000)`	Specifies the minimum number of VCs for the UBR service in percentage. The total values of `cwspMinVcCbr`, `cwspMinVcRtVbr`, `cwspMinVcNrtVbr`, `cwspMinVcAbr`, and `cwspMinVcUbr` cannot exceed 1000000, for example, 100%. The value of this variable is interpreted in the format of *xxx.xxx*. For example, *xxx.xxxx* is a value 750000 and is interpreted as 75.0000%. Max-Access: read-write Status: current	`0`
cwspMaxVcBwCbr	`Integer32 (0..1000000)`	Specifies the maximum bandwidth for a VC that is used for the real-time CBR service. The value is in units of cps. Max-Access: read-write Status: current	`0`

Table 5-37 Port CAC Configuration Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspMaxVcBwRtVbr	Integer32 (0..1000000)	Specifies the maximum bandwidth for a VC that is used for the real-time VBR service. The value is in units of cps. Max-Access: read-write Status: current	0
cwspMaxVcBwNrtVbr	Integer32 (0..1000000)	Specifies the maximum bandwidth for a VC that is used for the NRT VBR service. The value is in units of cps. Max-Access: read-write Status: current	0
cwspMaxVcBwAbr	Integer32 (0..1000000)	Specifies the maximum bandwidth for a VC that is used for the ABR service. The value is in units of cps. Max-Access: read-write Status: current	0
cwspMaxVcBwUbr	Integer32 (0..1000000)	Specifies the maximum bandwidth that is allowed for the UBR service. The value is in units of cps. Max-Access: read-write Status: current	0
cwspDefaultCdvtCbr	Integer32 (0..2147483647)	Specifies the default Cell Delay Variation Tolerance (CDVT) for the CBR service. Max-Access: read-write Status: current	1024
cwspDefaultCdvtRtVbr	Integer32 (0..2147483647)	Specifies the default CDVT for the real-time VBR service. Max-Access: read-write Status: current	1024
cwspDefaultCdvtNrtVbr	Integer32 (0..2147483647)	Specifies the default CDVT for the NRT VBR service. Max-Access: read-write Status: current	1024
cwspDefaultCdvtAbr	Integer32 (0..2147483647)	Specifies the default CDVT for the ABR service. Max-Access: read-write Status: current	1024

Table 5-37 Port CAC Configuration Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspDefaultCdvtUbr	`Integer32 (0..2147483647)`	Specifies the default CDVT for the UBR service. Max-Access: read-write Status: current	`1024`
cwspDefaultMbsRtVbr	`Integer32 (0..2147483647)`	Specifies the default Maximum Burst Size (MBS) for the real-time VBR service. Max-Access: read-write Status: current	`1024`
cwspDefaultMbsNrtVbr	`Integer32 (0..2147483647)`	Specifies the default MBS for the NRT VBR service. Max-Access: read-write Status: current	`1024`

Port Loading Table

The object identifier for each MIB object is listed in Table 5-38.

Table 5-38 Port Loading Table Object Identifiers

Name	Object Identifier
cwspLoadTable	::= { cwspLoad 1 }
cwspLoadEntry	::= { cwspLoadTable 1 }
cwspLoadBwTotal	::= { cwspLoadEntry 1 }
cwspLoadMaxBwCbr	::= { cwspLoadEntry 2 }
cwspLoadMaxBwRtVbr	::= { cwspLoadEntry 3 }
cwspLoadMaxBwNrtVbr	::= { cwspLoadEntry 4 }
cwspLoadMaxBwAbr	::= { cwspLoadEntry 5 }
cwspLoadMaxBwUbr	::= { cwspLoadEntry 6 }
cwspLoadBwAvail	::= { cwspLoadEntry 7 }
cwspLoadAvlBwCbr	::= { cwspLoadEntry 8 }
cwspLoadAvlBwRtVbr	::= { cwspLoadEntry 9 }
cwspLoadAvlBwNrtVbr	::= { cwspLoadEntry 10 }
cwspLoadAvlBwAbr	::= { cwspLoadEntry 11 }
cwspLoadAvlBwUbr	::= { cwspLoadEntry 12 }
cwspLoadVcAvail	::= { cwspLoadEntry 13 }
cwspLoadAvlVcCbr	::= { cwspLoadEntry 14 }
cwspLoadAvlVcRtVbr	::= { cwspLoadEntry 15 }
cwspLoadAvlVcNrtVbr	::= { cwspLoadEntry 16 }

Table 5-38 Port Loading Table Object Identifiers (continued)

Name	Object Identifier
cwspLoadAvlVcAbr	::= { cwspLoadEntry 17 }
cwspLoadAvlVcUbr	::= { cwspLoadEntry 18 }
cwspLoadCtdCbr	::= { cwspLoadEntry 19 }
cwspLoadCtdRtVbr	::= { cwspLoadEntry 20 }
cwspLoadCtdNrtVbr	::= { cwspLoadEntry 21 }
cwspLoadCtdAbr	::= { cwspLoadEntry 22 }
cwspLoadCtdUbr	::= { cwspLoadEntry 23 }
cwspLoadCdvCbr	::= { cwspLoadEntry 24 }
cwspLoadCdvRtVbr	::= { cwspLoadEntry 25 }
cwspLoadCdvNrtVbr	::= { cwspLoadEntry 26 }
cwspLoadCdvAbr	::= { cwspLoadEntry 27 }
cwspLoadCdvUbr	::= { cwspLoadEntry 28 }
cwspLoadClr0Cbr	::= { cwspLoadEntry 29 }
cwspLoadClr0RtVbr	::= { cwspLoadEntry 30 }
cwspLoadClr0NrtVbr	::= { cwspLoadEntry 31 }
cwspLoadClr0Abr	::= { cwspLoadEntry 32 }
cwspLoadClr0Ubr	::= { cwspLoadEntry 33 }
cwspLoadClr01Cbr	::= { cwspLoadEntry 34 }
cwspLoadClr01RtVbr	::= { cwspLoadEntry 35 }
cwspLoadClr01NrtVbr	::= { cwspLoadEntry 36 }
cwspLoadClr01Abr	::= { cwspLoadEntry 37 }
cwspLoadClr01Ubr	::= { cwspLoadEntry 38 }
cwspLoadMinGurCrCbr	::= { cwspLoadEntry 39 }
cwspLoadMinGurCrRtVbr	::= { cwspLoadEntry 40 }
cwspLoadMinGurCrNrtVbr	::= { cwspLoadEntry 41 }
cwspLoadMinGurCrAbr	::= { cwspLoadEntry 42 }
cwspLoadMinGurCrUbr	::= { cwspLoadEntry 43 }

The MIB objects are listed in Table 5-39.

Table 5-39 Port Loading Table MIB Objects

Name	Syntax	Description	Default Value
cwspLoadTable	`Sequence of CwspLoadEntry`	Specifies the load information for each interface. Max-Access: not-accessible Status: current	none
cwspLoadEntry	`CwspLoadEntry`	Contains an entry for the load information. Max-Access: not-accessible Status: current The index contains `ifIndex`.	none
cwspLoadBwTotal	`Integer32 (0..2147483647)`	Determines the total bandwidth of the interface. Max-Access: read-only Status: current	none
cwspLoadMaxBwCbr	`Integer32 (0..2147483647)`	Determines the maximum bandwidth for the CBR service. Max-Access: read-only Status: current	none
cwspLoadMaxBwRtVbr	`Integer32 (0..2147483647)`	Determines the maximum bandwidth for the real-time VBR service. Max-Access: read-only Status: current	none
cwspLoadMaxBwNrtVbr	`Integer32 (0..2147483647)`	Determines the maximum bandwidth for the NRT VBR service. Max-Access: read-only Status: current	none
cwspLoadMaxBwAbr	`Integer32 (0..2147483647)`	Determines the maximum bandwidth for the ABR service. Max-Access: read-only Status: current	none
cwspLoadMaxBwUbr	`Integer32 (0..2147483647)`	Determines the maximum bandwidth for the UBR service. Max-Access: read-only Status: current	none
cwspLoadBwAvail	`Integer32 (0..2147483647)`	Determines the total available bandwidth of the interface. Max-Access: read-only Status: current	none

Table 5-39 Port Loading Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspLoadAvlBwCbr	Integer32 (0..2147483647)	Determines the available bandwidth for the CBR service. Max-Access: read-only Status: current	none
cwspLoadAvlBwRtVbr	Integer32 (0..2147483647)	Determines the available bandwidth for the real-time VBR service. Max-Access: read-only Status: current	none
cwspLoadAvlBwNrtVbr	Integer32 (0..2147483647)	Determines the available bandwidth for the NRT VBR service. Max-Access: read-only Status: current	none
cwspLoadAvlBwAbr	Integer32 (0..2147483647)	Determines the available bandwidth for the ABR service. Max-Access: read-only Status: current	none
cwspLoadAvlBwUbr	Integer32 (0..2147483647)	Determines the available bandwidth for the UBR service. Max-Access: read-only Status: current	none
cwspLoadVcAvail	Integer32 (0..2147483647)	Determines the total number of available VCs for the interface. Max-Access: read-only Status: current	none
cwspLoadAvlVcCbr	Integer32 (0..2147483647)	Determines the number of VCs used by the CBR service. Max-Access: read-only Status: current	none
cwspLoadAvlVcRtVbr	Integer32 (0..2147483647)	Determines the number of VCs used by the real-time VBR service. Max-Access: read-only Status: current	none
cwspLoadAvlVcNrtVbr	Integer32 (0..2147483647)	Determines the number of VCs used by the NRT VBR service. Max-Access: read-only Status: current	none

Table 5-39 Port Loading Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspLoadAvlVcAbr	`Integer32 (0..2147483647)`	Determines the number of VCs used by the ABR service. Max-Access: read-only Status: current	none
cwspLoadAvlVcUbr	`Integer32 (0..2147483647)`	Determines the number of VCs used by the UBR service. Max-Access: read-only Status: current	none
cwspLoadCtdCbr	`Integer32 (0..2147483647)`	Specifies the cell transfer delay for the CBR service. Max-Access: read-only Status: current	none
cwspLoadCtdRtVbr	`Integer32 (0..2147483647)`	Specifies the cell transfer delay for the real-time VBR service. Max-Access: read-only Status: current	none
cwspLoadCtdNrtVbr	`Integer32 (0..2147483647)`	Specifies the cell transfer delay for the NRT VBR service. Max-Access: read-only Status: current	none
cwspLoadCtdAbr	`Integer32 (0..2147483647)`	Specifies the cell transfer delay for the ABR service. Max-Access: read-only Status: current	none
cwspLoadCtdUbr	`Integer32 (0..2147483647)`	Specifies the cell transfer delay for the UBR service. Max-Access: read-only Status: current	none
cwspLoadCdvCbr	`Integer32 (0..2147483647)`	Specifies the cell delay variation for the CBR service. Max-Access: read-only Status: current	none
cwspLoadCdvRtVbr	`Integer32 (0..2147483647)`	Specifies the cell delay variation for the real-time VBR service. Max-Access: read-only Status: current	none

Table 5-39 Port Loading Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspLoadCdvNrtVbr	`Integer32 (0..2147483647)`	Specifies the cell delay variation for the nonreal-time VBR service. Max-Access: read-only Status: current	none
cwspLoadCdvAbr	`Integer32 (0..2147483647)`	Specifies the cell delay variation for the ABR service. Max-Access: read-only Status: current	none
cwspLoadCdvUbr	`Integer32 (0..2147483647)`	Specifies the cell delay variation for the UBR service. Max-Access: read-only Status: current	none
cwspLoadClr0Cbr	`Integer32`	Specifies the cell loss ratio is called -0 for the CBR service. The value -1 means not applicable. Max-Access: read-only Status: current	none
cwspLoadClr0RtVbr	`Integer32`	Specifies the cell loss ratio is called -0 for the real-time VBR service. The value -1 means not applicable. Max-Access: read-only Status: current	none
cwspLoadClr0NrtVbr	`Integer32`	Specifies the cell loss ratio is called -0 for the nonreal-time VBR service. The value -1 means not applicable. Max-Access: read-only Status: current	none
cwspLoadClr0Abr	`Integer32`	Specifies the cell loss ratio is -0 for the ABR service. The value -1 means not applicable. Max-Access: read-only Status: current	none
cwspLoadClr0Ubr	`Integer32`	Specifies the cell loss ratio is called -0 for the UBR service. The value -1 means not applicable. Max-Access: read-only Status: current	none
cwspLoadClr01Cbr	`Integer32`	Specifies the cell loss ratio is called -1 for the CBR service. The value -1 means not applicable. Max-Access: read-only Status: current	none

Table 5-39 Port Loading Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspLoadClr01RtVbr	Integer32	Specifies the cell loss ratio is called -1 for the real-time VBR service. The value -1 means not applicable. Max-Access: read-only Status: current	none
cwspLoadClr01NrtVbr	Integer32	Specifies the cell loss ratio is called -1 for the NRT VBR service. The value -1 means not applicable. Max-Access: read-only Status: current	none
cwspLoadClr01Abr	Integer32	Specifies the cell loss ratio is called -1 for the ABR service. The value -1 means not applicable. Max-Access: read-only Status: current	none
cwspLoadClr01Ubr	Integer32	Specifies the cell loss ratio is called -1 for the UBR service. The value -1 means not applicable. Max-Access: read-only Status: current	none
cwspLoadMinGurCrCbr	Integer32 (0..2147483647)	Specifies the minimum guaranteed cell rate capacity for the CBR service. Max-Access: read-only Status: current	none
cwspLoadMinGurCrRtVbr	Integer32 (0..2147483647)	Specifies the minimum guaranteed cell rate capacity for the real-time VBR service. Max-Access: read-only Status: current	none
cwspLoadMinGurCrNrtVbr	Integer32 (0..2147483647)	Specifies the minimum guaranteed cell rate capacity for the NRT VBR service. Max-Access: read-only Status: current	none
cwspLoadMinGurCrAbr	Integer32 (0..2147483647)	Specifies the minimum guaranteed cell rate capacity for the ABR service. Max-Access: read-only Status: current	none
cwspLoadMinGurCrUbr	Integer32 (0..2147483647)	Specifies the minimum guaranteed cell rate capacity for the UBR service. Max-Access: read-only Status: current	none

Interface Operation Table

The object identifier for each MIB object is listed in Table 5-40.

Table 5-40 Interface Operation Table Object Identifiers

Name	Object Identifier
cwspOperationTable	::= { cwspOperation 1 }
cwspOperationEntry	::= { cwspOperationTable 1 }
cwspOperIlmiEnable	::= { cwspOperationEntry 1 }
cwspOperIfcType	::= { cwspOperationEntry 2 }
cwspOperIfcSide	::= { cwspOperationEntry 3 }
cwspOperMaxVPCs	::= { cwspOperationEntry 4 }
cwspOperMaxVCCs	::= { cwspOperationEntry 5 }
cwspOperMaxVpiBits	::= { cwspOperationEntry 6 }
cwspOperMaxVciBits	::= { cwspOperationEntry 7 }
cwspOperUniType	::= { cwspOperationEntry 8 }
cwspOperUniVersion	::= { cwspOperationEntry 9 }
cwspOperDeviceType	::= { cwspOperationEntry 10 }
cwspOperIlmiVersion	::= { cwspOperationEntry 11 }
cwspOperNniSigVersion	::= { cwspOperationEntry 12 }
cwspOperMaxSvpcVpi	::= { cwspOperationEntry 13 }
cwspOperMinSvpcVpi	::= { cwspOperationEntry 14 }
cwspOperMaxSvccVpi	::= { cwspOperationEntry 15 }
cwspOperMinSvccVpi	::= { cwspOperationEntry 16 }
cwspOperMaxSvccVci	::= { cwspOperationEntry 17 }
cwspOperMinSvccVci	::= { cwspOperationEntry 18 }
cwspOperAddrPlanSupported	::= { cwspOperationEntry 19 }

The MIB objects are listed in Table 5-41.

Table 5-41 Interface Operation Table MIB Objects

Name	Syntax	Description	Default Value
cwspOperationTable	`Sequence of CwspOperationEntry`	Specifies the interface operation table that contains the runtime negotiated values between the platform, PNNI controller, and peer for the interface. Max-Access: not-accessible Status: current	none
cwspOperationEntry	`CwspOperationEntry`	Specifies an entry in the interface operation table that contains the negotiated values for the interface. Max-Access: not-accessible Status: current The index contains `ifIndex`.	none
cwspOperIlmiEnable	`TruthValue`	Explains the operational state of Interim Local Management Interface (ILMI). Max-Access: read-only Status: current	none
cwspOperIfcType	`Integer { publicUni(1), privateUni(2), iisp(3), pnni(4), aini(5), enni(6)}`	Specifies the interface type. It contains the following: • Public User-Network Interface (publicUni) • Private User-Network Interface (privateUni) • Interim Inter-Switch Signaling Protocol (IISP) • Private Network-to-Network Interface (PNNI) • ATM Inter-Network Interface (AINI) • Enhanced Network-to-Network Interface (ENNI) Max-Access: read-only Status: current	none
cwspOperIfcSide	`INTEGER { userSide(1), networkSide(2), symmetric(3) }`	Explains the IME type of the ATM device that is determined by the automatic interface type determination procedure. Refer to *ATM Forum ILMI 4.0*, Section 8.3.4.1. Max-Access: read-only Status: current	none
cwspOperMaxVPCs	`Integer32`	Specifies the maximum number of switched and permanent VPCs supported on this ATM interface. Max-Access: read-only Status: current	none

Table 5-41 Interface Operation Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspOperMaxVCCs	`Integer32`	Specifies the maximum number of switched and permanent VCCs supported on this ATM interface. Max-Access: read-only Status: current	none
cwspOperMaxVpiBits	`Integer32`	Specifies the maximum number of active VPI bits on this ATM interface. Max-Access: read-only Status: current	none
cwspOperMaxVciBits	`Integer32`	Specifies the maximum number of active VCI bits on this ATM interface. Max-Access: read-only Status: current	none
cwspOperUniType	`INTEGER { public(1), private(2) }`	Specifies the type of the ATM device, for example, public or private. Max-Access: read-only Status: current	none
cwspOperUniVersion	`INTEGER { uni20(1), uni30(2), uni31(3), uni40(4), ituDss2(5) frf4(6) unsupported(7) ip(8)}`	Indicates the latest version of the *ATM Forum UNI Signaling Specification* that is supported on this ATM Interface. If this value is not present, a version of the UNI earlier than 3.1 is supported. If the peer IME value of this object is the same as or later than the local IME value, the version corresponding to the local IME value is attempted. Otherwise, if the peer IME value of this object is used earlier and supported locally, the local IME can use the version corresponding to the peer IME value. Otherwise, compatibility of the two IMEs cannot be assumed. Max-Access: read-only Status: current	none
cwspOperDeviceType	`INTEGER { user(1), node(2) }`	Determines the type of ATM device. This object is used in an automatic ATM Interface-Type determination procedure so that a correct operational ATM Interface-Type can be determined. An ATM End System can take the value `user(1)`, and an ATM network node can take the value `node(2)`. Max-Access: read-only Status: current	none

Table 5-41 Interface Operation Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspOperIlmiVersion	`INTEGER { unsupported(1), version4point0(2) }`	Indicates the latest version of the *ATM Forum ILMI Specification* that is supported on this ATM Interface. If the peer IME value of this object is the same as or later than the local IME value, the version corresponding to the local IME value is attempted. Otherwise, if the peer IME value of this object is earlier, and supported locally, the local IME can use the version that corresponds to the peer IME value. Otherwise, compatibility of the two IMEs cannot be assumed. If this object is not present, a version of the ILMI earlier than 4.0 is supported. Max-Access: read-only Status: current	none
cwspOperNniSig Version	`INTEGER{unsupported(1), iisp(2), pnniVersion1point0(3), enni(4), aini(5) }`	Indicates the latest version of the *ATM Forum PNNI Signaling Specification* supported on this ATM Interface. **Note** The PNNI routing version is not determined through ILMI. If the peer IME value of this object is the same as or later than the local IME value, the version that corresponds to the local IME value is attempted. Otherwise, if the peer IME value of this object is earlier and supported locally, the local IME can use the version that corresponds to the peer IME value. Otherwise, compatibility of the two IMEs cannot be assumed. **Note** `enni` is not a signaling protocol. Max-Access: read-only Status: current	none
cwspOperMaxSvpc Vpi	`Integer32`	Determines the maximum switched VPC VPI. Max-Access: read-only Status: current	none
cwspOperMinSvpc Vpi	`Integer32`	Determines the minimum switched VPC VPI. Max-Access: read-only Status: current	none
cwspOperMaxSvcc Vpi	`Integer32`	Determines the maximum switched VCC VPI. Max-Access: read-only Status: current	none
cwspOperMinSvcc Vpi	`Integer32`	Determines the minimum switched VCC VPI. Max-Access: read-only Status: current	none

Table 5-41 Interface Operation Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspOperMaxSvccVci	`Integer32`	Determines the maximum switched VCC VCI. Max-Access: read-only Status: current	none
cwspOperMinSvccVci	`Integer32`	Determines the minimum switched VCC VCI. Max-Access: read-only Status: current	none
cwspOperAddrPlanSupported	`Integer{both(1), aesa(2), e164(3) }`	Specifies the ATM address plan supported on a public UNI. For all other interfaces, the value is `aesa`. Max-Access: read-only Status: current	none

Port Signaling Statistics

The object identifier for each MIB object is listed in Table 5-42.

Table 5-42 Port Signaling Statistics Table Object Identifiers

Name	Object Identifier
cwspSigStatsTable	::= { cwspSigStats 1 }
cwspSigStatsEntry	::= { cwspSigStatsTable 1 }
cwspSigCounterReset	::= { cwspSigStatsEntry 1 }
cwspCallProcRcv	::= { cwspSigStatsEntry 2 }
cwspConnectRcv	::= { cwspSigStatsEntry 3 }
cwspConnectAckRcv	::= { cwspSigStatsEntry 4 }
cwspSetupRcv	::= { cwspSigStatsEntry 5 }
cwspReleaseRcv	::= { cwspSigStatsEntry 6 }
cwspReleaseComplRcv	::= { cwspSigStatsEntry 7 }
cwspRestartRcv	::= { cwspSigStatsEntry 8 }
cwspRestartAckRcv	::= { cwspSigStatsEntry 9 }
cwspStatusRcv	::= { cwspSigStatsEntry 10 }
cwspStatusEnqRcv	::= { cwspSigStatsEntry 11 }
cwspNotifyRcv	::= { cwspSigStatsEntry 12 }
cwspAlertRcv	::= { cwspSigStatsEntry 13 }
cwspProgressRcv	::= { cwspSigStatsEntry 14 }
cwspAddPtyRcv	::= { cwspSigStatsEntry 15 }
cwspAddPtyAckRcv	::= { cwspSigStatsEntry 16 }
cwspAddPtyRejRcv	::= { cwspSigStatsEntry 17 }

Table 5-42 Port Signaling Statistics Table Object Identifiers (continued)

Name	Object Identifier
cwspDropPtyRcv	::= { cwspSigStatsEntry 18 }
cwspIncorrectMsgRcv	::= { cwspSigStatsEntry 20 }
cwspTimerExpiries	::= { cwspSigStatsEntry 21 }
cwspLastCause	::= { cwspSigStatsEntry 22 }
cwspLastDiagnostic	::= { cwspSigStatsEntry 23 }
cwspCallProcXmt	::= { cwspSigStatsEntry 24 }
cwspConnectXmt	::= { cwspSigStatsEntry 25 }
cwspConnectAckXmt	::= { cwspSigStatsEntry 26 }
cwspSetupXmt	::= { cwspSigStatsEntry 27 }
cwspReleaseXmt	::= { cwspSigStatsEntry 28 }
cwspReleaseComplXmt	::= { cwspSigStatsEntry 29 }
cwspRestartXmt	::= { cwspSigStatsEntry 30 }
cwspRestartAckXmt	::= { cwspSigStatsEntry 31 }
cwspStatusXmt	::= { cwspSigStatsEntry 32 }
cwspStatusEnqXmt	::= { cwspSigStatsEntry 33 }
cwspNotifyXmt	::= { cwspSigStatsEntry 34 }
cwspAlertXmt	::= { cwspSigStatsEntry 35 }
cwspProgressXmt	::= { cwspSigStatsEntry 36 }
cwspAddPtyXmt	::= { cwspSigStatsEntry 37 }
cwspAddPtyAckXmt	::= { cwspSigStatsEntry 38 }
cwspAddPtyRejXmt	::= { cwspSigStatsEntry 39 }
cwspDropPtyXmt	::= { cwspSigStatsEntry 40 }
cwspSscopStatus	::= { cwspSigStatsEntry 42 }

The MIB objects are listed in Table 5-43.

Table 5-43 Port Signaling Statistics Table MIB Objects

Name	Syntax	Description	Default Value
cwspSigStatsTable	`Sequence of CwspSigStatsEntry`	Contains the signaling statistics counters. Max-Access: not-accessible Status: current	none
cwspSigStatsEntry	`CwspSigStatsEntry`	Contains the entry that is used for signaling statistics. Max-Access: not-accessible Status: current The index contains `ifIndex`.	none

Table 5-43 Port Signaling Statistics Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspSigCounterReset	INTEGER { noop(1), -- none of following reset(2) -- resetting}	Writes the value two to reset all the signaling counters. Max-Access: read-write Status: current	noop
cwspCallProcRcv	Counter32	Determines the number of CALL PROCEEDING messages received on this interface. Max-Access: read-only Status: current	none
cwspConnectRcv	Counter32	Determines the number of CONNECT messages that is received on this interface. Max-Access: read-only Status: current	none
cwspConnectAckRcv	Counter32	Determines the number of CONNECT ACK messages that is received on this interface. Max-Access: read-only Status: current	none
cwspSetupRcv	Counter32	Determines the number of SETUP messages received on this interface. Max-Access: read-only Status: current	none
cwspReleaseRcv	Counter32	Determines the number of RELEASE messages received on this interface. Max-Access: read-only Status: current	none
cwspReleaseComplRcv	Counter32	Determines the number of RELEASE COMPLETE messages received on this interface. Max-Access: read-only Status: current	none
cwspRestartRcv	Counter32	Determines the number of RESTART messages received on this interface. Max-Access: read-only Status: current	none
cwspRestartAckRcv	Counter32	Determines the number of RESTART ACK messages received on this interface. Max-Access: read-only Status: current	none

Table 5-43 Port Signaling Statistics Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspStatusRcv	Counter32	Determines the number of STATUS messages received on this interface. Max-Access: read-only Status: current	none
cwspStatusEnqRcv	Counter32	Determines the number of STATUS ENQUIRY messages received on this interface. Max-Access: read-only Status: current	none
cwspNotifyRcv	Counter32	Determines the number of NOTIFY messages received on this interface. Max-Access: read-only Status: current	none
cwspAlertRcv	Counter32	Determines the number of ALERT messages received on this interface. Max-Access: read-only Status: current	none
cwspProgressRcv	Counter32	Determines the number of PROGRESS messages received on this interface. Max-Access: read-only Status: current	none
cwspAddPtyRcv	Counter32	Determines the number of ADD PARTY messages received on this interface. Max-Access: read-only Status: current	none
cwspAddPtyAckRcv	Counter32	Determines the number of ADD PARTY ACK messages received on this interface. Max-Access: read-only Status: current	none
cwspAddPtyRejRcv	Counter32	Determines the number of ADD PARTY REJECT messages received on this interface. Max-Access: read-only Status: current	none
cwspDropPtyRcv	Counter32	Determines the number of DROP PARTY messages received on this interface. Max-Access: read-only Status: current	none

Table 5-43 Port Signaling Statistics Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspIncorrectMsgRcv	Counter32	Determines the number of INCORRECT messages received on this interface. Max-Access: read-only Status: current	none
cwspTimerExpiries	Counter32	Determines the number of timeouts that occurred on this interface. Max-Access: read-only Status: current	none
cwspLastCause	Counter32	Indicates the last cause of the release or crankback. Max-Access: read-only Status: current	none
cwspLastDiagnostic	Integer32	Indicates the last diagnostic of the release or crankback. Max-Access: read-only Status: current	none
cwspCallProcXmt	Counter32	Determines the number of CALL PROCEEDING messages transmitted from this interface. Max-Access: read-only Status: current	none
cwspConnectXmt	Counter32	Determines the number of CONNECT messages transmitted from this interface. Max-Access: read-only Status: current	none
cwspConnectAckXmt	Counter32	Determines the number of CONNECT ACK messages transmitted from this interface. Max-Access: read-only Status: current	none
cwspSetupXmt	Counter32	Determines the number of SETUP messages transmitted from this interface. Max-Access: read-only Status: current	none
cwspReleaseXmt	Counter32	Determines the number of RELEASE messages transmitted from this interface. Max-Access: read-only Status: current	none

Table 5-43 Port Signaling Statistics Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspReleaseComplXmt	Counter32	Determines the number of RELEASE COMPLETE messages transmitted from this interface. Max-Access: read-only Status: current	none
cwspRestartXmt	Counter32	Determines the number of RESTART messages transmitted from this interface. Max-Access: read-only Status: current	none
cwspRestartAckXmt	Counter32	Determines the number of RESTART ACK messages transmitted from this interface. Max-Access: read-only Status: current	none
cwspStatusXmt	Counter32	Determines the number of STATUS messages transmitted from this interface. Max-Access: read-only Status: current	none
cwspStatusEnqXmt	Counter32	Determines the number of STATUS ENQUIRY messages transmitted from this interface. Max-Access: read-only Status: current	none
cwspNotifyXmt	Counter32	Determines the number of NOTIFY messages transmitted from this interface. Max-Access: read-only Status: current	none
cwspAlertXmt	Counter32	Determines the number of ALERT messages transmitted from this interface. Max-Access: read-only Status: current	none
cwspProgressXmt	Counter32	Determines the number of PROGRESS messages transmitted from this interface. Max-Access: read-only Status: current	none
cwspAddPtyXmt	Counter32	Determines the number of ADD PARTY messages transmitted from this interface. Max-Access: read-only Status: current	none

Table 5-43 Port Signaling Statistics Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspAddPtyAckXmt	Counter32	Determines the number of ADD PARTY ACK messages transmitted from this interface. Max-Access: read-only Status: current	none
cwspAddPtyRejXmt	Counter32	Determines the number of ADD PARTY REJECT messages transmitted from this interface. Max-Access: read-only Status: current	none
cwspDropPtyXmt	Counter32	Determines the number of DROP PARTY messages transmitted from this interface. Max-Access: read-only Status: current	none
cwspSscopStatus	INTEGER { up(1), -- sscop link is established down(2) -- sscop link is tailed}	Specifies the Sscop link status on an NNI. This object is meaningful in conjunction with the ciscoWANSscopLinkChange trap. Max-Access: read-only Status: current	none

Port Connection Data Table

The object identifier for each MIB object is listed in Table 5-44.

Table 5-44 Port Connection Data Table Object Identifiers

Name	Object Identifier
cwspConnTraceTable	::= { cwspConnTrace 4 }
cwspConnTraceEntry	::= { cwspConnTraceTable 1 }
cwspConnTraceDataIndex	::= { cwspConnTraceEntry 1 }
cwspConnTraceNodeId	::= { cwspConnTraceEntry 2 }
cwspConnTraceEgressPortId	::= { cwspConnTraceEntry 3 }
cwspConnTraceEgressVpi	::= { cwspConnTraceEntry 4 }
cwspConnTraceEgressVci	::= { cwspConnTraceEntry 5 }
cwspConnTraceEgressCallRef	::= { cwspConnTraceEntry 6 }
cwspConnTraceEgressPhyPortId	::= { cwspConnTraceEntry 7 }
cwspConnTraceLastNode	::= { cwspConnTraceEntry 8 }

The MIB objects are listed in Table 5-45.

Table 5-45 Port Connection Data Table MIB Objects

Name	Syntax	Description	Default Value
cwspConnTraceTable	`Sequence of CwspConnTraceEntry`	Contains the objects that show the traversed node information in the existing SVC call. Max-Access: not-accessible Status: current	none
cwspConnTraceEntry	`CwspConnTraceEntry`	Specifies the conceptual row of the `cwspConnTraceTable` that is used to return information about all the nodes. The table rows belong to the connection which the search criteria corresponds to an instance of the `cwspConnTraceCntlTable`. Max-Access: not-accessible Status: current The indexes are as follows: • `cwspConnTraceIndex` • `cwspConnTraceDataIndex`	none
cwspConnTraceDataIndex	`Integer32`	Specifies an unique entry in the `cwspConnTraceTable` along with the `cwspConnTraceIndex`. Max-Access: not-accessible Status: current	none
cwspConnTraceNodeId	`PnniNodeId`	Represents 22 bytes for the `nodeId` in the traced connection that is used for this octet string. Max-Access: read-only Status: current	none
cwspConnTraceEgressPortId	`PnniPortId`	Represents 4 bytes for the logical port ID of the traversed node that is used for this object. When the value 0 is specified, the destination node for the trace is reached. Max-Access: read-only Status: current	none
cwspConnTraceEgressVpi	`Integer32`	Shows the egress port VPI value for the traced connection. Max-Access: read-only Status: current	none

Table 5-45 Port Connection Data Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspConnTraceEgressVci	`Integer32`	Shows the egress port VCI value for the traced connection. Max-Access: read-only Status: current	none
cwspConnTraceEgressCallRef	`Integer32`	Shows the egress port call reference. Max-Access: read-only Status: current	none
cwspConnTraceEgressPhyPortId	`OCTET STRING (SIZE(8))`	Shows the egress port physical port identifier for the traversed node. If this object is 0, which means the destination node for the traced connection has been reached, the meaning for the bytes are as follows: • 1st byte—Defines the flag that is used by the CLI to decode the rest of bytes. • 2nd byte—Defines the shelf. • 3rd and 4th bytes—Defines the slot. • 5th byte—Defines the subslot. • 6th and 7th bytes—Defines the port. • 8th byte—Defines the subport. Max-Access: read-only Status: current	none
cwspConnTraceLastNode	`TruthValue`	Indicates if the entry contains the information about the last node where the trace ends. If it is the last node, the value is either `true(1)` or `false(2)`. Max-Access: read-only Status: current	none

Port Connection Trace Control Table

The object identifier for each MIB object is listed in Table 5-46.

Table 5-46 Port Connection Trace Control Table Object Identifiers

Name	Object Identifier
cwspConnTraceCntlTable	::= { cwspConnTrace 3 }
cwspConnTraceCntlEntry	::= { cwspConnTraceCntlTable 1 }
cwspConnTraceIndex	::= { cwspConnTraceCntlEntry 1 }
cwspConnTraceifIndex	::= { cwspConnTraceCntlEntry 2 }

Table 5-46 Port Connection Trace Control Table Object Identifiers (continued)

Name	Object Identifier
cwspConnTraceSrcVpi	::= { cwspConnTraceCntlEntry 3 }
cwspConnTraceSrcVci	::= { cwspConnTraceCntlEntry 4 }
cwspConnTraceType	::= { cwspConnTraceCntlEntry 5 }
cwspConnTraceCallRef	::= { cwspConnTraceCntlEntry 6 }
cwspConnTraceLeafRef	::= { cwspConnTraceCntlEntry 7 }
cwspConnTraceDestVpi	::= { cwspConnTraceCntlEntry 8 }
cwspConnTraceDestVci	::= { cwspConnTraceCntlEntry 9 }
cwspConnTraceDestCallRef	::= { cwspConnTraceCntlEntry 10 }
cwspConnTraceResultStatus	::= { cwspConnTraceCntlEntry 11 }
cwspConnTraceQueryStatus	::= { cwspConnTraceCntlEntry 12 }

The MIB objects are listed in Table 5-47.

Table 5-47 Port Connection Trace Control Table MIB Objects

Name	Syntax	Description	Default Value
cwspConnTraceCntlTable	`Sequence of CwspConnTraceCntlEntry`	Contains the objects that control the creation of the connection trace for the existing SVC call. Max-Access: not-accessible Status: current	none
cwspConnTraceCntlEntry	`CwspConnTraceCntlEntry`	Controls the connection trace creation. The NMS must acquire the next available index to create a row. Also, NMS needs to provide all read-create objects in the same Protocol Data Unit (PDU) with `RowStatus` to be `CreateAndGo`. Max-Access: not-accessible Status: current The index contains `cwspConnTraceIndex`.	none
cwspConnTraceIndex	`Integer32 (1..2147483647)`	Specifies the index to a row that contains a connection trace. Max-Access: not-accessible Status: current	none
cwspConnTraceifIndex	`InterfaceIndex`	Indicates the port to a trace connection for the `ifIndex`. Max-Access: read-create Status: current	none

Table 5-47 Port Connection Trace Control Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspConnTraceSrcVpi	`Integer32 (0..4095)`	Shows the VPI value of the starting point on this interface. Max-Access: read-create Status: current	none
cwspConnTraceSrcVci	`Integer32 (0..65535)`	Shows the VCI value of the starting point on this interface. The value is in the range of 32 to 65535 `(0=SPVP)`. Max-Access: read-create Status: current	none
cwspConnTraceType	`INTEGER { p2p(1), p2mp(2) }`	Sets this object to either point-to-point or point-to multipoint to specify tracing a point-to-point or point-to multipoint connection. Max-Access: read-create Status: current	none
cwspConnTraceCallRef	`Integer32`	Shows the call reference value of the call on this interface. Max-Access: read-only Status: current	none
cwspConnTraceLeafRef	`Integer32 (0..65535)`	Shows the value of the leaf reference (`EndPointReference`) of the call on this interface. This value is used to support the point-to-multipoint call trace. For the point-to-point call, this value is set to 0 by the NMS. Max-Access: read-create Status: current	none
cwspConnTraceDestVpi	`Integer32`	Shows the endpoint VPI value of the call on this interface. Max-Access: read-only Status: current	none
cwspConnTraceDestVci	`Integer32`	Shows the endpoint VCI value of the call on this interface. Max-Access: read-only Status: current	none
cwspConnTraceDestCallRef	`Integer32`	Shows the endpoint call reference on this interface. Max-Access: read-only Status: current	none

Table 5-47 Port Connection Trace Control Table MIB Objects (continued)

Name	Syntax	Description	Default Value
cwspConnTraceResultStatus	`INTEGER { traceInProgress(1), traceCompleted(2), traceIncompleted(3), traceExceededLength(4), traceContRefused(5), traceLackResource(6)}`	Shows the result of tracing the call. NMS can get a positive result for this attribute, for example, `traceCompleted`, before querying the `cwspConnTraceTable`. Max-Access: read-only Status: current	none
cwspConnTraceQueryStatus	`RowStatus`	Manages rows in this table. However, only `CreateAndGo`, `NotInService`, `Active`, and `Destroy` are supported. NMS sets only the value to be `CreateAndGo` to startup the trace. To remove a row, NMS resets this value to `Destroy`. The managed device can either return `Active` or `NotInService`. Max-Access: read-create Status: current	none

Cisco WAN SVC Conformance and Compliance Statements

The information on conformance is specific to SNMPv2. Table 5-48 lists the conformance groups.

Table 5-48 Cisco WAN SVC MIB Conformance Groups

Name	Object Identifier
ciscoWANSvcMIBConformance	::= { ciscoWANSvcMIB 3 }
ciscoWANSvcMIBCompliances	::= { ciscoWANSvcMIBConformance 1 }
ciscoWANSvcMIBGroups	::= { ciscoWANSvcMIBConformance 2 }

Cisco WAN SVC Compliance Statements

The compliance statement for the entities is used to support the CISCO SVC MIB. The following are the mandatory groups:

- `cwsInfoGrp`
- `cwspConfigGrp`
- `cwspCacConfigGrp`
- `cwspCallStatsGrp`
- `cwspSigStatsGrp`
- `cwspCallGrp`
- `cwspAbrCallGrp`
- `cwspPrefixGrp`
- `cwspLoadGrp`
- `cwspAddressGrp`

- CwspConnTraceGrp
- cwspOperationGrp

The object identifier used for ciscoWANSvcMIBCompliance is listed in Table 5-49.

Table 5-49 Cisco WAN SVC MIB Compliance Object Identifier

Name	Object Identifier
ciscoWANSvcMIBCompliance	::= { ciscoWANSvcMIBCompliances 1 }

Table 5-50 describes the compliance groups.

Table 5-50 Cisco WAN SVC MIB Compliance Groups

Name	Description
cwspSigConfigGrp	Specifies that the signaling configuration group is optional.
cwspSscopConfigGrp	Specifies that the Service Specific Connection Oriented Protocol (SSCOP) configuration group is optional.
cwspSscopStatsGrp	Specifies that the SSCOP statistics group is optional.
cwspRegAddressGrp	Specifies that the registered ATM address group is optional.
cwspRoutingGrp	Specifies that the PNNI routing group is optional.
cwspSpvcGrp	Specifies that the SPVC group is optional.
cwspSpvcNodePrefix	Specifies that write access is currently not permitted. Max-Access: read-only
cwspMaxP2pCalls	Specifies that write access is currently not permitted. Max-Access: read-only

Cisco WAN SVC Units of Conformance

The object identifier for each MIB object is listed in Table 5-51.

Table 5-51 Cisco WAN SVC MIB Units of Conformance Object Identifiers

Name	Description
cwsInfoGrp	::= { ciscoWANSvcMIBGroups 1 }
cwspConfigGrp	::= { ciscoWANSvcMIBGroups 2 }
cwspCacConfigGrp	::= { ciscoWANSvcMIBGroups 3 }
cwspCallStatsGrp	::= { ciscoWANSvcMIBGroups 4 }
cwspSigStatsGrp	::= { ciscoWANSvcMIBGroups 5 }
cwspCallGrp	::= { ciscoWANSvcMIBGroups 6 }
cwspAbrCallGrp	::= { ciscoWANSvcMIBGroups 7 }
cwspPrefixGrp	::= { ciscoWANSvcMIBGroups 8 }
cwspLoadGrp	::= { ciscoWANSvcMIBGroups 9 }

Table 5-51 Cisco WAN SVC MIB Units of Conformance Object Identifiers (continued)

Name	Description
cwspAddressGrp	::= { ciscoWANSvcMIBGroups 10 }
cwspSigConfigGrp	::= { ciscoWANSvcMIBGroups 11 }
cwspSscopConfigGrp	::= { ciscoWANSvcMIBGroups 12 }
cwspSscopStatsGrp	::= { ciscoWANSvcMIBGroups 13 }
cwspRegAddressGrp	::= { ciscoWANSvcMIBGroups 14 }
c**wspRoutingGrp**	::= { ciscoWANSvcMIBGroups 15 }
cwspConnTraceGrp	::= { ciscoWANSvcMIBGroups 16 }
cwspOperationGrp	::= { ciscoWANSvcMIBGroups 17 }
cwspSpvcGrp	::= { ciscoWANSvcMIBGroups 18 }

The objects are listed in Table 5-52.

Table 5-52 Cisco WAN SVC MIB Objects Used for Units of Conformance

Name	Objects	Description	Default Value
cwsInfoGrp	cwsSwRevision, cwsControllerStatus, cwspPnniStndbyControllerStatus, cwspPnniControllerStatus, cwspPnniControllerPhySlot	Describes a collection of objects that provides device level information. Status: current	none
cwspConfigGrp	cwspAdminStatus, cwspOperStatus, cwspSvcBlocked, cwspSpvcBlocked, cwspIlmiAddrRegEnable, cwspIlmiAutoConfEnable, cwspIlmiServRegEnable, cwspPhyIdentifier, cwspSignallingVpi, cwspSignallingVci, cwspRoutingVpi, cwspRoutingVci, cwspMaxVpiBits, cwspMaxVciBits, cwspUniVersion, cwspNniVersion, cwspUniType, cwspSide, cwspMaxP2pCalls, cwspMaxP2mpRoots, cwspMaxP2mpLeafs, cwspMinSvccVpi, cwspMaxSvccVpi, cwspMinSvccVci, cwspMaxSvccVci, cwspMinSvpcVpi, cwspMaxSvpcVpi, cwspEnhancedIisp, cwspConfigTableRowStatus, cwspAddrPlanSupported	Describes a collection of objects that provides information about the controller interface operation and associated parameters. Status: current	none

Table 5-52 Cisco WAN SVC MIB Objects Used for Units of Conformance (continued)

Name	Objects	Description	Default Value
cwspCacConfigGrp	cwspUtilFactorCbr, cwspUtilFactorRtVbr, cwspUtilFactorNrtVbr, cwspUtilFactorAbr, cwspUtilFactorUbr, cwspMaxBwCbr, cwspMaxBwRtVbr, cwspMaxBwNrtVbr, cwspMaxBwAbr, cwspMaxBwUbr, cwspMinBwCbr,cwspMinBwRtVbr, cwspMinBwNrtVbr, cwspMinBwAbr, cwspMinBwUbr, cwspMaxVcCbr, cwspMaxVcRtVbr, cwspMaxVcNrtVbr, cwspMaxVcAbr, cwspMaxVcUbr, cwspMinVcCbr, cwspMinVcRtVbr, cwspMinVcNrtVbr, cwspMinVcAbr, cwspMinVcUbr, cwspMaxVcBwCbr, cwspMaxVcBwRtVbr, cwspMaxVcBwNrtVbr, cwspMaxVcBwAbr, cwspMaxVcBwUbr, cwspDefaultCdvtCbr, cwspDefaultCdvtRtVbr, cwspDefaultCdvtNrtVbr, cwspDefaultCdvtAbr, cwspDefaultCdvtUbr, cwspDefaultMbsRtVbr, cwspDefaultMbsNrtVbr	Describes a collection of objects that provides CAC information for an interface. Status: current	none
cwspCallStatsGrp	cwspCountReset, cwspInCallAttempts, cwspInCallEstabs, cwspInCallFailures, cwspInFilterFailures, cwspInRouteFailures, cwspInResrcFailures, cwspInTimerFailures, cwspInCrankbacks, cwspOutCallAttempts, cwspOutCallEstabs, cwspOutCallFailures, cwspOutFilterFailures, cwspOutRouteFailures, cwspOutResrcFailures, cwspOutTimerFailures, cwspOutCrankbacks	Determines a collection of objects that provides information about call statistics of an interface. Status: current	none

Table 5-52 Cisco WAN SVC MIB Objects Used for Units of Conformance (continued)

Name	Objects	Description	Default Value
cwspSigStatsGrp	cwspSigCounterReset, cwspCallProcRcv, cwspConnectRcv, cwspConnectAckRcv, cwspSetupRcv, cwspReleaseRcv, cwspReleaseComplRcv, cwspRestartRcv, cwspRestartAckRcv, cwspStatusRcv, cwspStatusEnqRcv, cwspNotifyRcv, cwspAlertRcv, cwspProgressRcv, cwspAddPtyRcv, cwspAddPtyAckRcv, cwspAddPtyRejRcv, cwspDropPtyRcv, cwspIncorrectMsgRcv, cwspTimerExpiries, cwspLastCause, cwspLastDiagnostic, cwspCallProcXmt, cwspConnectXmt, cwspConnectAckXmt, cwspSetupXmt, cwspReleaseXmt, cwspReleaseComplXmt, cwspRestartXmt, cwspRestartAckXmt, cwspStatusXmt, cwspStatusEnqXmt, cwspNotifyXmt, cwspAlertXmt, cwspProgressXmt, cwspAddPtyXmt, cwspAddPtyAckXmt, cwspAddPtyRejXmt, cwspDropPtyXmt, cwspSscopStatus	Describes a collection of objects that provides information about signaling statistics. Status: current	none

Table 5-52 Cisco WAN SVC MIB Objects Used for Units of Conformance (continued)

Name	Objects	Description	Default Value
cwspCallGrp	cwspCallCallRef, cwspCallCallingAddress, cwspCallCalledAddress, cwspCallCallingSubAddr, cwspCallCalledSubAddr, cwspCallOtherIntfIndex, cwspCallOtherIntfVpi, cwspCallOtherIntfVci, cwspCallOtherIntfCallRef, cwspCallType, cwspCallStartTime, cwspCallBearerClass, cwspCallServiceCategory, cwspCallCastType, cwspCallClipSusceptibility, cwspCallFwdConformance, cwspCallFwdPcr, cwspCallFwdScr, cwspCallFwdMbs, cwspCallFwdCdvt, cwspCallFwdFrameDiscard, cwspCallBwdConformance, cwspCallBwdPcr, cwspCallBwdScr, cwspCallBwdMbs, cwspCallBwdCdvt, cwspCallBwdFrameDiscard, cwspCallMaxCtd, cwspCallMaxFwdCdv, cwspCallMaxFwdClr, cwspCallMaxBwdCdv, cwspCallMaxBwdClr	Determines a collection of objects that provides information about the point-to-point or point-to-multipoint SVC/SPVC calls. This object also provides information for the associated parameters. Status: current	none

Table 5-52 Cisco WAN SVC MIB Objects Used for Units of Conformance (continued)

Name	Objects	Description	Default Value
cwspAbrCallGrp	cwspAbrCallCallRef, cwspAbrCallClgAddress, cwspAbrCallCldAddress, cwspAbrCallClgSubAddr, cwspAbrCallCldSubAddr, cwspAbrCallOtherIntfIndex, cwspAbrCallOtherIntfVpi, cwspAbrCallOtherIntfVci, cwspAbrCallOtherIntfCallRef, cwspAbrCallType, cwspAbrCallStartTime, cwspAbrCallBearerClass, cwspAbrCallServiceCategory, cwspAbrCallCastType, cwspAbrCallClipSusceptibility, cwspAbrCallFwdConformance, cwspAbrCallFwdPcr, cwspAbrCallFwdMcr, cwspAbrCallFwdIcr, cwspAbrCallFwdRif, cwspAbrCallFwdRdf, cwspAbrCallFwdTbe, cwspAbrCallFwdNrm, cwspAbrCallFwdTrm, cwspAbrCallFwdAdtf, cwspAbrCallFwdCdf, cwspAbrCallFwdFrameDiscard, cwspAbrCallBwdPcr, cwspAbrCallBwdMcr, cwspAbrCallBwdIcr, cwspAbrCallBwdRif, cwspAbrCallBwdRdf, cwspAbrCallBwdTbe, cwspAbrCallBwdNrm, cwspAbrCallBwdTrm, cwspAbrCallBwdAdtf, cwspAbrCallBwdCdf, cwspAbrCallBwdFrameDiscard, cwspAbrCallFrtt, cwspAbrCallMaxCtd, cwspAbrCallMaxFwdCdv, cwspAbrCallMaxFwdClr, cwspAbrCallMaxBwdCdv, cwspAbrCallMaxBwdClr	Determines a collection of objects that provides information about ABR calls and associated parameters of an interface. Status: current	none
cwspPrefixGrp	cwspPrefixRowStatus	Specifies the network prefix(es) that are used for the ILMI address registration. Status: current	none

Table 5-52 Cisco WAN SVC MIB Objects Used for Units of Conformance (continued)

Name	Objects	Description	Default Value
cwspLoadGrp	cwspLoadBwTotal, cwspLoadMaxBwCbr, cwspLoadMaxBwRtVbr, cwspLoadMaxBwNrtVbr, cwspLoadMaxBwAbr, cwspLoadMaxBwUbr, cwspLoadBwAvail, cwspLoadAvlBwCbr, cwspLoadAvlBwRtVbr, cwspLoadAvlBwNrtVbr, cwspLoadAvlBwAbr, cwspLoadAvlBwUbr, cwspLoadVcAvail, cwspLoadAvlVcCbr, cwspLoadAvlVcRtVbr, cwspLoadAvlVcNrtVbr, cwspLoadAvlVcAbr, cwspLoadAvlVcUbr, cwspLoadCtdCbr, cwspLoadCtdRtVbr, cwspLoadCtdNrtVbr, cwspLoadCtdAbr, cwspLoadCtdUbr, cwspLoadCdvCbr, cwspLoadCdvRtVbr, cwspLoadCdvNrtVbr, cwspLoadCdvAbr, cwspLoadCdvUbr, cwspLoadClr0Cbr, cwspLoadClr0RtVbr, cwspLoadClr0NrtVbr, cwspLoadClr0Abr, cwspLoadClr0Ubr, cwspLoadClr01Cbr, cwspLoadClr01RtVbr, cwspLoadClr01NrtVbr, cwspLoadClr01Abr, cwspLoadClr01Ubr, cwspLoadMinGurCrCbr, cwspLoadMinGurCrRtVbr, cwspLoadMinGurCrNrtVbr, cwspLoadMinGurCrAbr, cwspLoadMinGurCrUbr	Determines a collection of objects that provides information about load information and associated parameters of an interface. Status: current	none
cwspAddressGrp	cwspAddrType, cwspAddrProto, cwspAddrPlan, cwspAddrScope, cwspAddrRedistribute, cwspAddressRowStatus	Determines collection of objects that provides information about ATM End System Addresses. Status: current	none

Table 5-52 Cisco WAN SVC MIB Objects Used for Units of Conformance (continued)

Name	Objects	Description	Default Value
cwspSigConfigGrp	cwspSigCfgT301, cwspSigCfgT303, cwspSigCfgT308, cwspSigCfgT310, cwspSigCfgT316, cwspSigCfgT317, cwspSigCfgT322, cwspSigCfgT397, cwspSigCfgT398, cwspSigCfgT399	Determines a collection of objects that provides information about the signaling configuration and associated parameters. Status: current	none
cwspSscopConfigGrp	cwspSscopTmrCC, cwspSscopTmrKeepAlive, cwspSscopTmrNoResp, cwspSscopTmrPoll, cwspSscopTmtIdle, cwspSscopMaxCC, cwspSscopMaxPD, cwspSscopMaxSTAT	Determines a collection of objects about the SSCOP configuration and associated parameters. Status: current	none
cwspSscopStatsGrp	cwspSscopCounterReset, cwspSscopIgnoredPduRcv, cwspSscopBgnRcv, cwspSscopBgakRcv, cwspSscopEndRcv, cwspSscopEndakRcv, cwspSscopRsRcv, cwspSscopRsakRcv, cwspSscopBgrejRcv, cwspSscopSdRcv, cwspSscopSdpRcv, cwspSscopPollRcv, cwspSscopStatRcv, cwspSscopUstatRcv, cwspSscopUdRcv, cwspSscopMdRcv, cwspSscopErakRcv, cwspSscopBgnXmt, cwspSscopBgakXmt, cwspSscopEndXmt, cwspSscopEndakXmt, cwspSscopRsXmt, cwspSscopRsakXmt, cwspSscopBgrejXmt, cwspSscopSdXmt, cwspSscopSdpXmt, cwspSscopPollXmt, cwspSscopStatXmt, cwspSscopUstatXmt, cwspSscopUdXmt, cwspSscopMdXmt, cwspSscopErakXmt	Determines a collection of objects that provides information about the SSCOP statistics. Status: current	none
cwspRegAddressGrp	cwspRegAtmAddress, cwspRegAddressOrgSscope	Determines a collection of objects that provides information about the registered ATM address. Status: current	none

Table 5-52 Cisco WAN SVC MIB Objects Used for Units of Conformance (continued)

Name	Objects	Description	Default Value
cwspRoutingGrp	ciscoWANPnniLinkStatus, ciscoWANPnniPglStatus, ciscoWANPnniReachability, ciscoWANPnniRemoteNodeId, ciscoWANPnniPortId	Determines a collection of objects that provides information about the PNNI topology. Status: current	none
cwspConnTraceGrp	cwspConnTraceAvail, cwspConnTraceNextIndex, cwspConnTraceifIndex, cwspConnTraceSrcVpi, cwspConnTraceSrcVci, cwspConnTraceType, cwspConnTraceCallRef, cwspConnTraceLeafRef, cwspConnTraceDestVpi, cwspConnTraceDestVci, cwspConnTraceDestCallRef, cwspConnTraceResultStatus, cwspConnTraceQueryStatus, cwspConnTraceNodeId, cwspConnTraceEgressPortId, cwspConnTraceEgressVpi, cwspConnTraceEgressVci, cwspConnTraceEgressCallRef, cwspConnTraceEgressPhyPortId cwspConnTraceLastNode	Determines the value is true if this entry is the last node. If this entry is not the last node, the value is false. Status: current	none
cwspOperationGrp	cwspOperIlmiEnable, cwspOperIfcType, cwspOperIfcSide, cwspOperMaxVPCs, cwspOperMaxVCCs, cwspOperMaxVpiBits, cwspOperMaxVciBits, cwspOperUniType, cwspOperUniVersion, cwspOperDeviceType, cwspOperIlmiVersion, cwspOperNniSigVersion, cwspOperMaxSvpcVpi, cwspOperMinSvpcVpi, cwspOperMaxSvccVpi, cwspOperMinSvccVpi, cwspOperMaxSvccVci, cwspOperMinSvccVci, cwspOperFailReason, cwspOperAddrPlanSupported, cwspIlmiSecureLink, cwspIlmiAttachmentPoint, cwspIlmiLocalAttrStd, cwspIlmiUCSMEnable	Determines a collection of objects that provides information about the runtime negotiated values between the platform, PNNI controller, and peer on an interface. Status: current	none
cwspSpvcGrp	ciscoWANSpvcFailReason, cwspSpvcNodePrefix	Determines a collection of objects that provides SPVC related information in the PNNI controller. Status: current	none

CHAPTER 6

RPM MIB Objects

This chapter describes the MIB objects used for Router Processor Module (RPM).

Contents of this chapter include:

- Cisco WAN RPM Subinterface MIB Objects
- Cisco RPM Connection Extension MIB Objects

Cisco WAN RPM Subinterface MIB Objects

This section describes the individual MIB objects that make up the Cisco WAN RPM Subinterface file, which resides in the CISCO-WAN-RPM-SUBIF-MIB.my file. These MIBs are used to provision backplane subinterfaces on RPM.

Note The Cisco WAN RPM Subinterface MIB is defined under the StrataCom Enterprise.

The object identifier for each MIB object is listed in Table 6-1.

Table 6-1 Cisco WAN RPM Subinterface Object Identifiers

Name	Object Identifier
cwRpmSubIfMIBObjects	::= { ciscoWanRpmSubIfMIB 1 }
cwRpmSubIf	::= { cwRpmSubIfMIBObjects 1 }
cwRpmSubIfTable	::= { cwRpmSubIf 1 }
cwRpmSubIfEntry	::= { cwRpmSubIfTable 1 }
cwrSubIfSlotNum	::= { cwRpmSubIfEntry 1 }
cwrSubIfNum	::= { cwRpmSubIfEntry 2 }
cwrSubIfAdapterIf	::= { cwRpmSubIfEntry 3 }
cwrSubIfType	::= { cwRpmSubIfEntry 4 }
cwrSubIfIpAddress	::= { cwRpmSubIfEntry 5 }
cwrSubIfSubnetMask	::= { cwRpmSubIfEntry 6 }
cwrSubIfRowStatus	::= { cwRpmSubIfEntry 7 }

The MIB objects are listed in Table 6-2.

Table 6-2 Cisco WAN RPM Subinterface MIB Objects

Name	Syntax	Description	Default Value
cwRpmSubIfTable	`SEQUENCE OF CwRpmSubIfEntry`	Contains the RPM backplane subinterfaces. Max-Access: not-accessible Status: current	none
cwRpmSubIfEntry	`CwRpmSubIfEntry`	Corresponds to a RPM backplane subinterface for each entry. It contains the RPM backplane subinterface number, IP address, type, and state. Max-Access: not-accessible Status: current The indexes include: • `cwrSubIfSlotNum` • `cwrSubIfNum`	none
cwrSubIfSlotNum	`Unsigned32 (1..32)`	Specifies the slot number of the RPM card. Max-Access: not-accessible Status: current	none
cwrSubIfNum	`Unsigned32 (1..4294967295)`	Specifies the subinterface number. Max-Access: not-accessible Status: current	none
cwrSubIfAdapterIf	`Unsigned32`	Maps to the RPM for port adapter interface. Currently, one port adapter interface is supported. **Note** This object is supported as read-only. Max-Access: read-create Status: current	none
cwrSubIfType	`INTEGER { pointToPoint(1), multiPoint(2), labelSwitching(3)}`	Specifies the link type of this subinterface. Max-Access: read-create Status: current	none
cwrSubIfIpAddress	`IpAddress`	Specifies the 4-octet IP address of the RPM subinterface. Max-Access: read-create Status: current	none

Table 6-2 Cisco WAN RPM Subinterface MIB Objects (continued)

Name	Syntax	Description	Default Value
cwrSubIfSubnetMask	IpAddress	Specifies the 4-octet subnet mask of the RPM subinterface. Max-Access: read-create Status: current	none
cwrSubIfRowStatus	RowStatus	Specifies the administrative status of this subinterface. A row is created using the CreateAndGo option. When the row is successfully created, the RowStatus is set to active by the agent. When the subinterface is administratively downed, the RowStatus is set to notInService. If the subinterface is administratively upped again, the RowStatus is set to active again. The subinterface is deleted by setting the RowStatus to destroy. Other options such as CreateAndWait is not used. Max-Access: read-create Status: current	none

Cisco WAN RPM Subinterface Conformance and Compliance Statements

The object identifiers are listed in Table 6-3 for conformance.

Table 6-3 Cisco WAN RPM Subinterface Object Identifier

Name	Object Identifier
ciscoWanRpmSubIfMIBConformance	::= { ciscoWanRpmSubIfMIB 2 }
ciscoWanRpmSubIfMIBCompliances	::= { ciscoWanRpmSubIfMIBConformance 1 }
ciscoWanRpmSubIfMIBGroups	::= { ciscoWanRpmSubIfMIBConformance 2 }

Cisco WAN RPM Subinterface Compliance Statement

The compliance object identifier is listed in Table 6-4.

Table 6-4 Cisco WAN RPM Subinterface Object Identifier

Name	Object Identifier
ciscoWanRpmSubIfMIBCompliance	::= { ciscoWanRpmSubIfMIBCompliances 1 }

One MIB object is listed in Table 6-5.

Table 6-5 Cisco WAN RPM Subinterface Compliance MIB Object

Name	Mandatory Group	Description	Default Value
ciscoWanRpmSubIfMIB Compliance	`ciscoWanRpmSubIfMIBGroup`	Describes the compliance statement for the Cisco MGX RPM subinterface management group. Status: current	none

Cisco WAN RPM Subinterface Units of Conformance

One object identifier is listed in Table 6-6.

Table 6-6 Cisco WAN RPM Subinterface Units of Conformance Object Identifier

Name	Object Identifier
ciscoWanRpmSubIfMIBGroup	::= { ciscoWanRpmSubIfMIBGroups 1 }

One object is listed in Table 6-7 for units of conformance.

Table 6-7 Cisco WAN RPM Subinterface Units of Conformance MIB Object

Name	Objects	Description	Default Value
ciscoWanRpmSubIfMIB Group	`cwrSubIfAdapterIf, cwrSubIfType, cwrSubIfIpAddress, cwrSubIfSubnetMask, cwrSubIfRowStatus}`	Describes the conformance for the Cisco MGX RPM subinterface management group. Status: current	none

Cisco RPM Connection Extension MIB Objects

This section describes the individual MIB objects that make up the Cisco RPM Connection Extension file, which resides in the CISCO-WAN-RPM-CONN-EXT-MIB.my file. These MIBs are used to provision the Permanent Virtual Circuit (PVC) endpoints and Switched Permanent Virtual Circuit (SPVC) on RPM.

Note The Cisco RPM Connection Extension MIB is defined under the StrataCom Enterprise.

The object identifier for each MIB object is listed in Table 6-8.

Table 6-8 Cisco RPM Connection Extension Object Identifiers

Name	Object Identifier
cwRpmConnExtMIBObjects	::= { ciscoWanRpmConnExtMIB 1 }
cwRpmConnExt	::= { cwRpmConnExtMIBObjects 1 }

Table 6-8 Cisco RPM Connection Extension Object Identifiers (continued)

Name	Object Identifier
cwRpmChanExtTable	::= { cwRpmConnExt 1 }
cwRpmChanExtEntry	::= { cwRpmChanExtTable 1 }
cwrChanSubInterface	::= { cwRpmChanExtEntry 1 }
cwrChanVcd	::= { cwRpmChanExtEntry 2 }
cwrChanAalEncapType	::= { cwRpmChanExtEntry 3 }
cwrChanVirtualTemplate	::= { cwRpmChanExtEntry 4 }
cwrChanInArpInterval	::= { cwRpmChanExtEntry 5 }
cwrChanOamLoopbkTxInterval	::= { cwRpmChanExtEntry 6 }
cwrChanOamManage	::= { cwRpmChanExtEntry 7 }
cwrChanOamRetryUpCount	::= { cwRpmChanExtEntry 8 }
cwrChanOamRetryDownCount	::= { cwRpmChanExtEntry 9 }
cwrChanOamRetryInterval	::= { cwRpmChanExtEntry 10 }

The MIB objects are listed in Table 6-9.

Table 6-9 Cisco RPM Connection Extension MIB Objects

Name	Syntax	Description	Default Value
cwRpmChanExtTable	`SEQUENCE OF CwRpmChanExtEntry`	Supports the provisioning of PVC endpoints and SPVCs on an RPM along with the `cwAtmChanCnfgTable`. `cwRpmChanExtTable` contains objects that are specific to RPM. This table is an augmentation to the `cwAtmChanCnfgTable`. Max-Access: not-accessible Status: current	none
cwRpmChanExtEntry	`CwRpmChanExtEntry`	Contains RPM-specific parameters for a connection endpoint on a RPM. Max-Access: not-accessible Status: current Augment: `cwAtmChanCnfgEntry`	none
cwrChanSubInterface	`Unsigned32 (0..4294967295)`	Specifies the RPM backplane subinterface where this channel is configured. Subinterface is 0 for the switch port. Max-Access: read-create Status: current	`0`
cwrChanVcd	`Unsigned32 (0..4095)`	Describes the Virtual Circuit Descriptor (VCD) used to identify a connection for the unique number. Max-Access: read-only Status: current	none

Table 6-9 Cisco RPM Connection Extension MIB Objects (continued)

Name	Syntax	Description	Default Value
cwrChanAalEncapType	INTEGER { aal5ciscoPPP(1), aal5muxAPOLLO(2), aal5muxAPPLETALK(3), aal5muxDECNET(4), aal5muxIP(5), aal5muxIPX(6), aal5muxPPP(7), aal5muxVINES(8), aal5muxXNS(9), aal5nlpid(10), aal5snap(11), ilmi(12), qsaal(13)}	Specifies the ATM Adaptation Layer (AAL) and encapsulation. Max-Access: read-create Status: current The parameters include: • aal5ciscoPPP—Specifies that cwrChanVirtualTemplate is required to be set for Cisco Point-to-Point (PPP) over ATM. • aal5muxAPOLLO—Specifies the Multiplexing Device (mux) type virtual circuit that is used for the Apollo protocol. • aal5muxAPPLETALK—Specifies the mux type for the virtual circuit that is used for the AppleTalk protocol. • aal5muxDECNET—Specifies the mux type for the virtual circuit that is used for the DECnet protocol. • aal5muxIP—Specifies the mux type for the virtual circuit that is used for the IP protocol. • aal5muxIPX—Specifies the mux type for the virtual circuit that is used for the IPX protocol. • aal5muxPPP—Specifies the mux type for the virtual circuit that is used for PPP. cwrChanVirtualTemplate is required to be set for this type. • aal5muxVINES—Specifies the mux type for the virtual circuit that is used for Virtual Integrated Network Service (VINES). • aal5muxXNS—Specifies the mux type for the virtual circuit that is used for Xerox Network Systems (XNS). • aal5nlpid—Allows ATM interfaces to interoperate with High-Speed Serial Interface (HSSI). • aal5snap—Supports Inverse ARP (Address Resolution Protocol) that is used only for encapsulation. Logical Link Control/Subnetwork Access Protocol (LLC/SNAP) precedes the protocol datagram. • ilmi—Sets up communication with Interim Local Management Interface (ILMI). • qsaal—Sets up or tears down the Switched Virtual Circuit (SVC) for the Permanent Virtual Circuit (PVC) signaling type.	aal5snap

Table 6-9 Cisco RPM Connection Extension MIB Objects (continued)

Name	Syntax	Description	Default Value
cwrChanVirtual Template	`Unsigned32 (0..25)`	Requires Cisco PPP or mux-type PPP encapsulation, for example, `cwrChanAalEncapType = aal5ciscoPPP` or `aal5muxPPP`. `cwrChanVirtualTemplate` specifies the number used to identify the virtual template. If the virtual template is not used, this value is 0. Max-Access: read-create Status: current	`0`
cwrChanInArpInterval	`Unsigned32 (0..60)`	Specifies how often Inverse ARP datagrams are sent on this channel. The value 0 disables `InArp` for this channel. This object is supported only for `aal5snap` encapsulation, for example, `cwrChanAalEncapType = aal5snap`. Units: minutes Max-Access: read-create Status: current	`15`
cwrChanOamLoopbk TxInterval	`Unsigned32 (0..600)`	Specifies how often to generate an Operation, Administration, Maintenance (OAM) F5 loopback cell from this channel. The value 0 disables generation of OAM F5 loopback cells. Units: seconds Max-Access: read-create Status: current	`10`
cwrChanOamManage	`TruthValue`	Enables and disables OAM management for this channel. If set to `true(1)`, OAM management is enabled. Max-Access: read-create Status: current	`false`
cwrChanOamRetryUp Count	`Unsigned32 (1..600)`	Determines the number of consecutive end-to-end F5 OAM loopback cell responses that must be received to change a PVC connection state to `up`. This object applies only if OAM management is enabled, for example, `cwrChanOAMManage = true`. Max-Access: read-create Status: current	`3`

Table 6-9 Cisco RPM Connection Extension MIB Objects (continued)

Name	Syntax	Description	Default Value
cwrChanOamRetry DownCount	`Unsigned32 (1..600)`	Determines the number of consecutive end-to-end F5 OAM loopback cell responses that are not received to change a PVC connection state to `down`. This object applies only if OAM management is enabled, for example, `cwrChanOAMManage = true`. Max-Access: read-create Status: current	`5`
cwrChanOamRetry Interval	`Unsigned32 (1..1000)`	Specifies the frequency for end-to-end F5 OAM loopback cells that are transmitted when a change in up/down state of a PVC is being verified. This object applies only if OAM management is enabled, for example, `cwrChanOAMManage = true`. Units: seconds Max-Access: read-create Status: current	`1`

Cisco RPM Connection Extension Conformance and Compliance Statements

The object identifiers are listed in Table 6-10 for conformance.

Table 6-10 Cisco RPM Connection Extension Conformance Groups

Name	Object Identifier
ciscoWanRpmConnExtMIBConformance	::= { ciscoWanRpmConnExtMIB 2 }
ciscoWanRpmConnExtMIBCompliances	::= { ciscoWanRpmConnExtMIBConformance 1 }
ciscoWanRpmConnExtMIBGroups	::= { ciscoWanRpmConnExtMIBConformance 2 }

Cisco RPM Connection Extension Compliance Statements

The compliance object identifier is listed in Table 6-11.

Table 6-11 Cisco RPM Connection Extension Object Identifier

Name	Object Identifier
ciscoWanRpmConnExtMIBCompliance	::= { ciscoWanRpmConnExtMIBCompliances 1 }

One MIB object is listed in Table 6-12.

Table 6-12 Cisco RPM Connection Extension Compliance MIB Object

Name	Mandatory Group	Description	Default Value
ciscoWanRpmConn ExtMIBCompliance	ciscoWanRpmConnExtMIBGroup	Describes the compliance statement for the RPM connection management group.	none

Cisco RPM Connection Extension Units of Conformance

One object identifier is listed in Table 6-13.

Table 6-13 Cisco RPM Connection Extension Units of Conformance Object Identifier

Name	Object Identifier
ciscoWanRpmConnExtMIBGroup	::= { ciscoWanRpmConnExtMIBGroups 1 }

One object is listed in Table 6-14 for units of conformance.

Table 6-14 Cisco RPM Connection Extension Compliance MIB Object

Name	Objects	Description	Default Value
ciscoWanRpmConn ExtMIBGroup	cwrChanSubInterface, cwrChanVcd, cwrChanAalEncapType, cwrChanVirtualTemplate, cwrChanInArpInterval, cwrChanOamLoopbkTxInterval, cwrChanOamManage, cwrChanOamRetryUpCount, cwrChanOamRetryDownCount, cwrChanOamRetryInterval}	Describes the conformance for the RPM connection management group. Status: current	none

CHAPTER 7

Traps and Alarms Supported

This chapter describes the traps and alarms used for AXSM, PXM45, PNNI, and RPM.

Contents of this chapter include:

- Overview
- Traps Reference
- AXSM Traps and Alarms
- PXM Traps and Alarms
- PNNI Traps and Alarms
- RPM Traps
- Mandatory Trap Varbinds
- Trap-Specific Varbinds

Each section on a service module trap provides the trap name, trap number, trap specific-variables (varbinds), and a description.

Note All mandatory trap varbinds are described in Table 7-13. For information about the miscellaneous trap varbinds, see Table 7-14.

Overview

Traps are messages used to inform the SNMP manager of a condition on the network. An agent can send unsolicited traps to the manager to notify the manager of network conditions. Traps indicate improper user authentication, restarts, up or down link status, TCP connection closure, or other significant events.

Traps Reference

Table 7-1 lists all the traps in numeric order, type of service module, and location.

Table 7-1 Traps Reference List

Name	Number	Location	Service Module
cwChassisIntegratedAlarm	60001	For more information, see “PXM Traps and Alarms” section of this chapter.	PXM45
cwUserLogin	60002	For more information, see “PXM Traps and Alarms” section of this chapter.	PXM45
cwUserLogout	60003	For more information, see “PXM Traps and Alarms” section of this chapter.	PXM45
cwShelfRestart	60004	For more information, see “PXM Traps and Alarms” section of this chapter.	PXM45
cwNodeNameChange	60006	For more information, see “PXM Traps and Alarms” section of this chapter.	PXM45
cwIpAddressChange	60007	For more information, see “PXM Traps and Alarms” section of this chapter.	PXM45
cwControllerAdd	60008	For more information, see “PXM Traps and Alarms” section of this chapter.	PXM45
cwControllerDelete	60009	For more information, see “PXM Traps and Alarms” section of this chapter.	PXM45
cwDiskDbIntegCheckDone	60024	For more information, see “PXM Traps and Alarms” section of this chapter.	PXM45
cwDiskDbIntegCheckFailed	60025	For more information, see “PXM Traps and Alarms” section of this chapter.	PXM45
cwChassisTemperatureNormal	60026	For more information, see “PXM Traps and Alarms” section of this chapter.	PXM45
cwChassisTemperatureAboveNormal	60027	For more information, see “PXM Traps and Alarms” section of this chapter.	PXM45
cwChassisDclevelNormal	60028	For more information, see “PXM Traps and Alarms” section of this chapter.	PXM45
cwChassisDclevelBelowNormal	60029	For more information, see “PXM Traps and Alarms” section of this chapter.	PXM45
cwChassisPowersupplyNormal	60030	For more information, see “PXM Traps and Alarms” section of this chapter.	PXM45
cwChassisPowersupplyFailed	60031	For more information, see “PXM Traps and Alarms” section of this chapter.	PXM45
cwChassisFanrpmNormal	60032	For more information, see “PXM Traps and Alarms” section of this chapter.	PXM45
cwChassisFanrpmBelowNormal	60033	For more information, see “PXM Traps and Alarms” section of this chapter.	PXM45

Table 7-1 Traps Reference List (continued)

Name	Number	Location	Service Module
cwChassisDclevelAboveNormal	60034	For more information, see "PXM Traps and Alarms" section of this chapter.	PXM45
cwModuleInserted	60051	For more information, see "PXM Traps and Alarms" section of this chapter.	PXM45
cwModuleRemoved	60052	For more information, see "PXM Traps and Alarms" section of this chapter.	PXM45
cwModuleMismatch	60053	For more information, see "PXM Traps and Alarms" section of this chapter.	PXM45
cwModuleActive	60055	For more information, see "PXM Traps and Alarms" section of this chapter.	PXM45
cwModuleStandby	60056	For more information, see "PXM Traps and Alarms" section of this chapter.	PXM45
cwModuleFailed	60057	For more information, see "PXM Traps and Alarms" section of this chapter.	PXM45
cwLineModuleInserted	60058	For more information, see "PXM Traps and Alarms" section of this chapter.	PXM45
cwLineModuleRemoved	60059	For more information, see "PXM Traps and Alarms" section of this chapter.	PXM45
cwLineModuleMismatch	60060	For more information, see "PXM Traps and Alarms" section of this chapter.	PXM45
cwLineModuleMismatchClear	60062	For more information, see "PXM Traps and Alarms" section of this chapter.	PXM45
cwCoreCardSwitch	60078	For more information, see "PXM Traps and Alarms" section of this chapter.	PXM45
cwRedundancyAdd	60079	For more information, see "PXM Traps and Alarms" section of this chapter.	PXM45
cwRedundancyDelete	60080	For more information, see "PXM Traps and Alarms" section of this chapter.	PXM45
cwRedundancyActivate	60081	For more information, see "PXM Traps and Alarms" section of this chapter.	PXM45
cwRedundancyRevert	60082	For more information, see "PXM Traps and Alarms" section of this chapter.	PXM45
cwSonetLineUp	60101	For more information, see "SONET Line" section of this chapter.	AXSM
cwSonetLineDown	60102	For more information, see "SONET Line" section of this chapter.	AXSM
cwSonetLineConfigChange	60103	For more information, see "SONET Line" section of this chapter.	AXSM
cwSonetLineInAlarm	60104	For more information, see "SONET Line" section of this chapter.	AXSM

Table 7-1 Traps Reference List (continued)

Name	Number	Location	Service Module
cwSonetLineNoAlarm	60105	For more information, see “SONET Line” section of this chapter.	AXSM
cwSonetLineLpbkEnable	60106	For more information, see “SONET Line” section of this chapter.	AXSM
cwSonetLineLpbkDisable	60107	For more information, see “SONET Line” section of this chapter.	AXSM
cwSonetSectionStatAlarm	60108	For more information, see “SONET Line” section of this chapter.	AXSM
cwSonetLineStatAlarm	60109	For more information, see “SONET Line” section of this chapter.	AXSM
cwSonetPathStatAlarm	60110	For more information, see“SONET Line” section of this chapter.	AXSM
cwSonetApsEnable	60121	For more information, see “SONET Line” section of this chapter.	AXSM
cwSonetApsDisable	60122	For more information, see “SONET Line” section of this chapter.	AXSM
cwSonetApsLineFailure	60123	For more information, see “SONET Line” section of this chapter.	AXSM
cwSonetApsLineSwitch	60124	For more information, see “SONET Line” section of this chapter.	AXSM
cwSonetApsLineSwitchFailure	60125	For more information, see “SONET Line” section of this chapter.	AXSM
cwSonetApsRedundantLineInAlarm	60126	For more information, see “SONET Line” section of this chapter.	AXSM
cwSonetApsRedundantLineClear	60127	For more information, see “SONET Line” section of this chapter.	AXSM
cwSonetApsFailureClear	60128	For more information, see “SONET Line”section of this chapter.	AXSM
cwSonetApsConfigChange	60129	For more information, see “SONET Line”section of this chapter.	AXSM
cwSonetApsModeMismatch	60130	For more information, see “SONET Line” section of this chapter.	AXSM
cwSonetApsModeMismatchClear	60131	For more information, see “SONET Line” section of this chapter.	AXSM
cwDs3LineUp	60151	For more information, see “DS3 Line” section of this chapter.	AXSM
cwDs3LineDown	60152	For more information, see “DS3 Line” section of this chapter.	AXSM
cwDs3LineConfigChange	60153	For more information, see “DS3 Line” section of this chapter.	AXSM

Table 7-1 Traps Reference List (continued)

Name	Number	Location	Service Module
cwDs3LineInAlarm	60154	For more information, see “DS3 Line” section of this chapter.	AXSM
cwDs3LineNoAlarm	60155	For more information, see “DS3 Line” section of this chapter.	AXSM
cwDs3LineLpbkEnable	60156	For more information, see “DS3 Line” section of this chapter.	AXSM
cwDs3LineLpbkDisable	60157	For more information, see “DS3 Line” section of this chapter.	AXSM
cwDs3StatAlarm	60158	For more information, see “DS3 Line” section of this chapter.	AXSM
cwDs3PlcpInAlarm	60159	For more information, see “DS3 Line” section of this chapter.	AXSM
cwDs3PlcpNoAlarm	60160	For more information, see “DS3 Line” section of this chapter.	AXSM
cwDs3PlcpStatInAlarm	60161	For more information, see “DS3 Line” section of this chapter.	AXSM
cwaChanAdd	60301	For more information, see “Connection-Related Traps” section of this chapter.	AXSM RPM
cwaChanDelete	60302	For more information, see “Connection-Related Traps” section of this chapter.	AXSM RPM
cwaChanActive	60303	For more information, see “Connection-Related Traps” section of this chapter.	AXSM RPM
cwaChanFail	60304	For more information, see “Connection-Related Traps” section of this chapter.	AXSM RPM
cwaChanConfigChange	60305	For more information, see “Connection-Related Traps” section of this chapter.	AXSM RPM
cwaChanMultipleChanInAlarms	60306	For more information, see “Connection-Related Traps” section of this chapter.	AXSM RPM
cwaChanDown	60307	For more information, see “Connection-Related Traps” section of this chapter.	AXSM RPM
cwaChanUp	60308	For more information, see “Connection-Related Traps” section of this chapter.	AXSM RPM
cwaChanMajorAlarm	60309	For more information, see “Connection-Related Traps” section of this chapter.	AXSM
cwaChanMinorAlarm	60310	For more information, see “Connection-Related Traps” section of this chapter.	AXSM
cwAtmIfAdd	60351	For more information, see “Virtual Interface-Related Traps” section of this chapter.	AXSM
cwAtmIfDelete	60352	For more information, see “Virtual Interface-Related Traps” section of this chapter.	AXSM

Table 7-1 Traps Reference List (continued)

Name	Number	Location	Service Module
cwAtmIfUp	60353	For more information, see "Virtual Interface-Related Traps" section of this chapter.	AXSM
cwAtmIfDown	60354	For more information, see "Virtual Interface-Related Traps" section of this chapter.	AXSM
cwAtmIfConfigChange	60355	For more information, see "Virtual Interface-Related Traps" section of this chapter.	AXSM
cwAtmIfSctFileAlarm	60356	For more information, see "Virtual Interface-Related Traps" section of this chapter.	AXSM
cwAtmIfSctFileNoAlarm	60357	For more information, see "Virtual Interface-Related Traps" section of this chapter.	AXSM
cwCardIngSctFileAlarm	60358	For more information, see "Virtual Interface-Related Traps" section of this chapter.	AXSM
cwCardIngSctFileNoAlarm	60359	For more information, see "Virtual Interface-Related Traps" section of this chapter.	AXSM
cwCardIngSctFileIdChange	60360	For more information, see "Virtual Interface-Related Traps" section of this chapter.	AXSM
cwAtmPhyInAlarm	60371	For more information, see "ATM Line-Related Traps" section of this chapter.	AXSM
cwAtmPhyNoAlarm	60372	For more information, see "ATM Line-Related Traps" section of this chapter.	AXSM
cwAtmPhyConfigChange	60373	For more information, see "ATM Line-Related Traps" section of this chapter.	AXSM
cwRsrcPartAdd	60381	For more information, see "Resource Partition-Related Traps"section of this chapter.	AXSM RPM
cwRsrcPartDelete	60382	For more information, see "Resource Partition-Related Traps" section of this chapter.	AXSM RPM
cwRsrcPartConfigChange	60383	For more information, see "Resource Partition-Related Traps" section of this chapter.	AXSM RPM
cwRpmSubIfAdd	60401	For more information, see "RPM Subinterface-Related Traps" section of this chapter.	RPM
cwRpmSubIfDelete	60402	For more information, see "RPM Subinterface-Related Traps" section of this chapter.	RPM
cwRpmSubIfUp	60403	For more information, see "RPM Subinterface-Related Traps" section of this chapter.	RPM
cwRpmSubIfDown	60404	For more information, see "RPM Subinterface-Related Traps" section of this chapter.	RPM
cwRpmSubIfConfigChange	60405	For more information, see "RPM Subinterface-Related Traps" section of this chapter.	RPM
cwRpmSubIfOperUp	60406	For more information, see "RPM Subinterface-Related Traps" section of this chapter	RPM

Table 7-1 Traps Reference List (continued)

Name	Number	Location	Service Module
cwRpmSubIfOperDown	60407	For more information, see "RPM Subinterface-Related Traps" section of this chapter	RPM
cwRpmSubIfOperUnknown	60408	For more information, see "RPM Subinterface-Related Traps" section of this chapter	RPM
cwTrapsLost	60900	For more information, see "PXM Traps and Alarms" section of this chapter.	PXM1 PXM45 AXSM AXSM-E
cwBulkFileCreationStarted	60901	For more information, see "PXM Traps and Alarms" section of this chapter.	PXM45
cwBulkFileCreationDone	60902	For more information, see "PXM Traps and Alarms" section of this chapter.	PXM45
cwBulkFileCreationAborted	60903	For more information, see "PXM Traps and Alarms" section of this chapter.	PXM45
cwBulkNoTaskAvailable	60904	For more information, see "PXM Traps and Alarms" section of this chapter.	PXM45
cwBulkFileCreationFailed	60905	For more information, see "PXM Traps and Alarms" section of this chapter.	PXM45
cwStatScmIpAddressChange	60920	For more information, see "PXM Traps and Alarms" section of this chapter.	PXM45
cwStatFileInfo	60921	For more information, see "PXM Traps and Alarms" section of this chapter.	PXM45
cwFeederAdded	61001	For more information, see "Feeder-Related Traps" section of this chapter.	AXSM
cwFeederDeleted	61002	For more information, see "Feeder-Related Traps" section of this chapter.	AXSM
cwFeederLMIUp	61003	For more information, see "Feeder-Related Traps" section of this chapter.	AXSM
cwFeederLMIDown	61004	For more information, see "Feeder-Related Traps" section of this chapter.	AXSM
cwFeederConfigChange	61005	For more information, see "Feeder-Related Traps" section of this chapter.	AXSM
cwLMIAdded	61006	For more information, see "Local Management Interface-Related Traps" section of this chapter.	AXSM
cwLMIDeleted	61007	For more information, see "Local Management Interface-Related Traps" section of this chapter.	AXSM
cwLMIUp	61008	For more information, see "Local Management Interface-Related Traps" section of this chapter.	AXSM
cwlMIDown	61009	For more information, see "Local Management Interface-Related Traps" section of this chapter.	AXSM

Table 7-1 Traps Reference List (continued)

Name	Number	Location	Service Module
cwLMIConfigChange	61010	For more information, see "Local Management Interface-Related Traps" section of this chapter.	AXSM
ciscoWANSscopLinkChange	70003	For more information, see "PNNI Traps and Alarms" section of this chapter.	PXM45
ciscoWANControllerStateChange	70004	For more information, see "PNNI Traps and Alarms" section of this chapter.	PXM45
ciscoWANIntfAddTrap	70005	For more information, see "PNNI Traps and Alarms" section of this chapter.	PXM45
ciscoWANIntfDeleteTrap	70006	For more information, see "PNNI Traps and Alarms" section of this chapter.	PXM45
ciscoWANIntfConfigurtaionChangeTrap	70007	For more information, see "PNNI Traps and Alarms" section of this chapter.	PXM45
ciscoWANIntfOperationChangeTrap	70008	For more information, see "PNNI Traps and Alarms" section of this chapter.	PXM45
ciscoWANSpvcFailTrap	70009	For more information, see "PNNI Traps and Alarms" section of this chapter.	PXM45
ciscoWANPnniControllerStndbyFailed	70010	For more information, see "PNNI Traps and Alarms" section of this chapter.	PXM45
ciscoWANSpvcNodePrefixChange	70011	For more information, see "PNNI Traps and Alarms" section of this chapter.	PXM45
ciscoWANSpvcFailureTrap	70012	For more information, see "PNNI Traps and Alarms" section of this chapter.	PXM45
ciscoWANClockChangeTrap	70013	For more information, see "PNNI Traps and Alarms" section of this chapter.	PXM45

AXSM Traps and Alarms

Traps are generated by the AXSM card on various actions. Examples of these actions are adding, deleting, or modifying resources and line failures.

The following are the three types of traps:

- Line-related traps include traps arising from the SONET, DS3, or ATM lines.
- Virtual interface-related traps are associated with ATM port addition, modification, and deletion.
- Connection-related traps are associated with channel addition, modification, and deletion.

AXSM traps and alarms include:

- Line-Related Traps
- Virtual Interface-Related Traps
- Resource Partition-Related Traps
- Connection-Related Traps
- Local Management Interface-Related Traps

- Feeder-Related Traps

Line-Related Traps

Line-related traps include traps created from fault conditions on the SONET, DS3, or ATM lines.

SONET Line

SONET line-related traps are presented in Table 7-2.

Table 7-2 SONET Line-Related Traps

Name	Number	Specific Varbinds	Description
cwSonetLineUp	60101	`ifIndex, ifName`	This trap is sent when a SONET line is enabled.
cwSonetLineDown	60102	`ifIndex, ifName`	This trap is sent when a SONET line is disabled.
cwSonetLineConfigChange	60103	`ifIndex, ifName`	This trap is sent when a SONET line goes through any configuration change.
cwSonetLineInAlarm	60104	`ifIndex, ifName, sonetSectionCurrentStatus, sonetLineCurrentStatus, sonetPathCurrentStatus, ssTraceFailure, cspTraceFailure`	This trap is sent when the SONET line goes into alarm. The alarm status bitmap is one of the varbinds of the trap.
cwSonetLineNoAlarm	60105	`ifIndex, ifName, sonetSectionCurrentStatus, sonetLineCurrentStatus, sonetPathCurrentStatus, cssTraceFailure, cspTraceFailure`	This trap is sent when a SONET line alarm is cleared.
cwSonetLineLpbkEnable	60106	`ifIndex, ifName, sonetSectionCurrentStatus, sonetLineCurrentStatus, sonetPathCurrentStatus`	This trap is sent when loopback is enabled on a SONET line.
cwSonetLineLpbkDisable	60107	`ifIndex, ifName, sonetSectionCurrentStatus, sonetLineCurrentStatus, sonetPathCurrentStatus`	This trap is sent when loopback is disabled on SONET line.
cwSonetSectionStatAlarm	60108	`ifIndex, ifName, cwsSectionStatAlarmStatus`	This trap is sent when a SONET section layer encounters a statistical alarm.
cwSonetLineStatAlarm	60109	`ifIndex, ifName, cwsLineStatAlarmStatus`	This trap is sent when a SONET line layer encounters a statistical alarm.
cwSonetPathStatAlarm	60110	`ifIndex, ifName, cwsPathStatAlarmStatus`	This trap is sent when a SONET path layer encounters a statistical alarm.
cwSonetApsEnable	60121	`cwTrapIndex, csApsProtectionIndex, csApsActiveLine, ifName`	This trap is sent when the APS feature is enabled on a SONET line.
cwSonetApsDisable	60122	`cwTrapIndex, csApsProtectionIndex, csApsActiveLine, ifName`	This trap is sent when APS feature is disabled on a SONET line.

Table 7-2 SONET Line-Related Traps (continued)

Name	Number	Specific Varbinds	Description
cwSonetApsLineFailure	60123	cwTrapIndex, csApsProtectionIndex, csApsActiveLine, csApsLineFailureCode, ifName	This trap is sent when the SONET APS line encounters an abnormal condition.
cwSonetApsLineSwitch	60124	cwTrapIndex, csApsProtectionIndex, csApsActiveLine, csApsLineSwitchReason, ifName	This trap is sent when the SONET line APS switch occurs.
cwSonetApsLineSwitch Failure	60125	cwTrapIndex, csApsProtectionIndex, csApsActiveLine, csApsLineSwitchReason, ifName	This trap is sent when an abnormal condition occurs during the SONET APS switch.
cwSonetApsRedundantLine InAlarm	60126	cwTrapIndex, csApsProtectionIndex, csApsActiveLine, sonetSectionCurrentStatus, sonetLineCurrentStatus, sonetPathCurrentStatus, ifName	This trap is sent when the SONET APS redundant line is not in service.
cwSonetApsRedundantLine Clear	60127	cwTrapIndex, csApsProtectionIndex, csApsActiveLine, sonetSectionCurrentStatus, sonetLineCurrentStatus, sonetPathCurrentStatus, ifName	This trap is sent when the SONET APS redundant line alarm has been cleared.
cwSonetApsFailureClear	60128	cwTrapIndex, csApsProtectionIndex, csApsActiveLine, csApsLineFailureCode, ifName	This trap is sent when the SONET APS line failure has been cleared.
cwSonetApsConfigChange	60129	cwTrapIndex, csApsProtectionIndex, csApsActiveLine, ifName	This trap is sent when the SONET APS configuration parameters changes, for example, if signal degrade parameter changes.
cwSonetApsModeMismatch	60130	cwTrapIndex, csApsProtectionIndex, csApsActiveLine, ifName	This trap is sent when the SONET APS architecture mode on the two ends of the APS pair is different. This mismatch does not cause an APS failure; the mode on one end is automatically changed to match the other end.
cwSonetApsModeMismatch Clear	60131	cwTrapIndex, csApsProtectionIndex, csApsActiveLine, ifName	This trap is sent when the SONET APS architecture mode mismatch is cleared.

DS3 Line

DS3 Line-related traps are presented in Table 7-3.

Table 7-3 DS3 Line Related Traps

Name	Number	Specific Varbinds	Description
cwDs3LineUp	60151	ifIndex, ifName	This trap is sent when a DS3 line is enabled.
cwDs3LineDown	60152	ifIndex, ifName	This trap is sent when a DS3 line is disabled.
cwDs3LineConfigChange	60153	ifIndex, ifName	This trap is sent when a DS3 line goes through any configuration change.
cwDs3LineInAlarm	60154	ifIndex, ifName, dsx3LineStatus	This trap is sent when the DS3 line goes into alarm. The alarm status bitmap is one of the varbinds of the trap.
cwDs3LineNoAlarm	60155	ifIndex, ifName, dsx3LineStatus	This trap is sent when a DS3 line alarm is cleared.
cwDs3LineLpbkEnable	60156	ifIndex, ifName, dsx3LineStatus	This trap is sent when loopback is enabled on a DS3 line.
cwDs3LineLpbkDisable	60157	ifIndex, ifName, dsx3LineStatus	This trap is sent when loopback is disabled on SONET line.
cwDs3StatAlarm	60158	ifIndex, ifName, cds3LineStatisticalAlarmState	This trap is sent when a DS3 line encounters a statistical alarm.
cwDs3PlcpInAlarm	60159	ifIndex, ifName, cds3PlcpLineAlarmState	This trap is sent when a DS3 Plcp line encounters an alarm.
cwDs3PlcpNoAlarm	60160	ifIndex, ifName, cds3PlcpLineAlarmState	This trap is sent when a DS3 Plcp alarm is cleared.
cwDs3PlcpStatInAlarm	60161	ifIndex, ifName, cds3PlcpLineStatisticalAlarmState	This trap is sent when a DS3 Plcp line encounters a statistical alarm.

ATM Line-Related Traps

ATM line-related traps are presented in Table 7-4.

Table 7-4 ATM Line-Related Traps

Name	Number	Specific-Varbinds	Description
cwAtmPhyInAlarm	60371	ifIndex, ifName	This trap is sent when physical ATM cell layer loss of cell delineation occured.
cwAtmPhyNoAlarm	60372	ifIndex, ifName	This trap is sent when physical ATM cell layer loss of cell delineation alarm cleared.
cwAtmPhyConfigChange	60373	ifIndex, ifName	This trap is sent when ATM cell layer configuration changes.

Virtual Interface-Related Traps

The virtual interface-related traps are presented in Table 7-5.

Table 7-5 Virtual Interface-Related Traps

Name	Number	Specific Varbinds	Description
cwAtmIfAdd	60351	`ifIndex, ifName, caviFileId`	This trap is sent when an ATM port is deleted.
cwAtmIfDelete	60352	`ifIndex, ifName`	This trap is sent when an ATM port is deleted.
cwAtmIfUp	60353	`ifIndex, ifName`	This trap is sent when an ATM port is enabled.
cwAtmIfDown	60354	`ifIndex, ifName`	This trap is sent when an ATM port is disabled.
cwAtmIfConfigChange	60355	`ifIndex, ifName`	This trap is sent when an ATM port configuration changes.
cwAtmIfSctFileAlarm	60356	`ifIndex, ifName, caviFileId`	This trap is sent when the SCT file associated with an ATM port is detected to be missing or corrupted.
cwAtmIfSctFileNoAlarm	60357	`ifIndex, ifName`	This trap is sent when the ATM port comes out of the failed state because of a corrupt or missing SCT file.
cwCardIngSctFileAlarm	60358	`ifIndex, cwmIngressSCTFileId`	This trap is sent when the ingress SCT file for a card is detected as missing or corrupted on a switch over.
cwCardIngSctFileNoAlarm	60359	`ifIndex, cwmIngressSCTFileId`	Card ingress SCT file alarm cleared.
cwCardIngSctFileIdChange	60360	`ifIndex, cwmIngressSCTFileId`	Card ingress SCT file id is changed.

Resource Partition-Related Traps

The resource partition-related traps are listed in Table 7-6.

Note The resource partition-related traps are also supported by RPM.

Table 7-6 Resource Partition-Related Traps

Name	Number	Specific Varbinds	Description
cwRsrcPartAdd	60381	`ifIndex`, `ifName`, `cwTrapIndex`, `cwRsrcPartController`	This trap is sent when a resource partition is added on the logical port by the `ifindex` object. The `cwRsrcPartVpiLo` object contains the `cwRsrcPartID` value. The `cwTrapIndex` object contains the `cwRsrcPartController` value.
cwRsrcPartDelete	60382	`ifIndex`, `ifName`, `cwTrapIndex`, `cwRsrcPartController`	This trap is sent when a resource partition is deleted by the logical port by the `ifindex` object. The `cwRsrcPartVpiLo` object contains the `cwRsrcPartID` value. The `cwTrapIndex` object contains the `cwRsrcPartController` value.
cwRsrcPartConfig Change	60383	`ifIndex`, `ifName`, `cwTrapIndex`, `cwRsrcPartController`	This trap is sent when the configuration for the resource partition is changed. The `cwRsrcPartVpiLo` object contains the `cwRsrcPartID` value. The `cwTrapIndex` object contains the `cwRsrcPartController` value. The following are the modified objects: • `cwRsrcPartEgrMinBw` • `cwRsrcPartEgrMaxBw` • `cwRsrcPartIngMinBw` • `cwRsrcPartIngMaxBw` • `cwRsrcPartVpiLo` • `cwRsrcPartVpiHigh` • `cwRsrcPartVciLo` • `cwRsrcPartVciHigh` • `cwRsrcPartMinCon` • `cwRsrcPartMaxCon`

Connection-Related Traps

Connection-related traps are described in Table 7-7.

Note The applicable connection-related traps are also supported by RPM.

Table 7-7 Connection-Related Traps

Name	Number	Specific Varbinds	Description
cwaChanAdd[1]	60301	`ifIndex, ifName, cwaChanLocalVpi, cwaChanLocalVci, cwaChanVpcFlag, cwaChanIdentifier, cwaChanUploadCounter`	This trap is sent when either a Virtual Path Connection (VPC) or Virtual Channel Connection (VCC) gets added. The `cwaChanLocalVpi` object contains a value for `cwaChanVpi`. The `cwaChanLocalVci` object contains a value for `cwaChanVci`.
cwaChanDelete[1]	60302	`ifIndex, ifName, cwaChanLocalVpi, cwaChanLocalVci, cwaChanVpcFlag, cwaChanIdentifier, cwaChanUploadCounter`	This trap is sent when a channel gets deleted. The `cwaChanLocalVpi` object contains a value for `cwaChanVpi`. The `cwaChanLocalVci` object contains a value for `cwaChanVci`.
cwaChanActive[1]	60303	`ifIndex, ifName, cwaChanLocalVpi, cwaChanLocalVci, cwaChanVpcFlag, cwaChanIdentifier, cwaChanUploadCounter`	This trap is sent when a channel gets added or modified. The `cwaChanLocalVpi` object contains a value for `cwaChanVpi`. The `cwaChanLocalVci` object contains a value for `cwaChanVci`.
cwaChanFail[1]	60304	`ifIndex, ifName, cwaChanLocalVpi, cwaChanLocalVci, cwaChanVpcFlag, cwaChanIdentifier, cwaChanAlarmState, cwaChanEgressXmtState, cwaChanEgressRcvState, cwaChanIngressXmtState, cwaChanIngressRcvState`	This trap is sent when there is a channel failure. The `cwaChanLocalVpi` object contains a value for `cwaChanVpi`. The `cwaChanLocalVci` object contains a value for `cwaChanVci`.
cwaChanConfig Change[1]	60305	`ifIndex, ifName, cwaChanLocalVpi, cwaChanLocalVci, cwaChanVpcFlag, cwaChanIdentifier, cwaChanUploadCounter`	This trap is sent when channel parameters are changed. The `cwaChanLocalVpi` object contains a value for `cwaChanVpi`. The `cwaChanLocalVci` object contains a value for `cwaChanVci`.

Table 7-7 Connection-Related Traps (continued)

Name	Number	Specific Varbinds	Description
cwaChanMultiple ChanInAlarms[1]	60306	none	This trap is sent when many channels are in an alarm state. The `cwaChanMultipleChanInAlarms` trap replaces the failed traps for each connection. The `cwaChanMultipleChanInAlarms` trap is sent during a regular interval as long as the alarm condition exists. For example, if Cisco WAN Manager does not receive another trap in the next minute, it uploads a connection alarm file to the database to synchronize with the condition of the node or connection.

cwaChanDown[1]	60307	`ifIndex,`
		`ifName,`
		`cwaChanLocalVpi,`
		`cwaChanLocalVci,`
		`cwaChanVpcFlag,`
		`cwaChanIdentifier,`
		`cwaChanUploadCounter`
			This trap is sent when a channel is down.

cwaChanUp[1]	60308	`ifIndex,`
		`ifName,`
		`cwaChanLocalVpi,`
		`cwaChanLocalVci,`
		`cwaChanVpcFlag,`
		`cwaChanIdentifier,`
		`cwaChanUploadCounter,`
		`cwaChanOperStatus`
			This trap indicates that a VP/VC is administratively up.
			The `cwaChanLocalVpi` object contains `cwaChanVpi` value. The `cwaChanLocalVci` object contains a `cwaChanVci` value.

Table 7-7 Connection-Related Traps (continued)

Name	Number	Specific Varbinds	Description
cwaChanMajor Alarm	60309	`ifIndex, ifName, cwaChanLocalVpi, cwaChanLocalVci, cwaChanVpcFlag, cwaChanIdentifier, cwaChanAlarmState, cwaChanOperStatus, cwaChanEgressXmtState, cwaChanEgressRcvState, cwaChanIngressXmtState, cwaChanIngressRcvState`	This trap indicates a connection failure. The indications of a `cwaChanMajorAlarm` include: • The endpoint is conditioned, for example, derouted. • Call control (CC) failure. • Mismatch in the database configuration and hardware configuration. The `cwaChanAlarmState` object in the `cwaChanMajorAlarm` trap contains a bitmap representation of all possible alarm conditions on VCC/VPC. When the `cwaChanMajorAlarm` trap notification is sent, the `cwaChanOperStatus` is set to `oper_status=FAIL(2)`. The `cwaChanLocalVpi` object contains the `cwaChanVpi` value. The `cwaChanLocalVci` object contains the `cwaChanVci` value.
cwaChanMinor Alarm	60310	`ifIndex, ifName, cwaChanLocalVpi, cwaChanLocalVci, cwaChanVpcFlag, cwaChanIdentifier, cwaChanAlarmState, cwaChanOperStatus, cwaChanEgressXmtState, cwaChanEgressRcvState, cwaChanIngressXmtState, cwaChanIngressRcvState`	This trap indicates a failure outside the network and is reflected in `oper_status=OK`. When the `cwaChanMinorAlarm` trap notification is sent, the `cwaChanOperStatus` is set to `operOk(1)`. The indications of a `cwaChanMinorAlarm` include: • Detects Ingress Alarm Indication Signal (AIS). • Detects Egress Alarm Indication Signal (AIS). • Detects Ingress A-bit alarm. The `cwaChanAlarmState` object in the `cwaChanMinorAlarm` trap contains a bitmap representation of all possible alarm conditions on VCC/VPC. The `cwaChanLocalVpi` object contains the `cwaChanVpi` value. The `cwaChanLocalVci` object contains the `cwaChanVci` value.

1. This connection-related trap is also supported by RPM.

Local Management Interface-Related Traps

The Local Management Interface (LMI) traps are listed in Table 7-8.

Table 7-8 Local Management Interface-Related Traps

Name	Number	Specific-Varbinds	Description
cwLMIAdded	61006	cwTrapIndex, cwfLMIType	Configures the new atmVirtualInterface port as XLMI. **Note** cwTrapIndex is the interface index.
cwLMIDeleted	61007	cwTrapIndex, cwfLMIType	Configures the previous atmVirtualInterface port to run XLMI. **Note** cwTrapIndex is the interface index.
cwLMIUp	61008	cwTrapIndex, cwfLMIType	Determines that the LMI connection is UP.
cwlMIDown	61009	cwTrapIndex, cwfLMIType	Determines that the LMI connection is DOWN.
cwLMIConfigChange	61010	cwTrapIndex, cwfLMIType	Determines that the IP address, node name, and node status changes.

Feeder-Related Traps

The feeder-related traps are listed in Table 7-9.

Table 7-9 Feeder-Related Traps

Name	Number	Specific-Varbinds	Description
cwFeederAdded	61001	cwTrapIndex	This trap is sent when a feeder is added on a service module. The cwTrapIndex object contains the value of cwfFeederIfNum in the cwfFeederTable.
cwFeederDeleted	61002	cwTrapIndex	This trap is sent when the feeder is deleted on service module. The cwTrapIndex object contains the value of cwfFeederIfNum in the cwfFeederTable.
cwFeederLMIUp	61003	cwTrapIndex	This trap is sent when the LMI connection is established after the feeder is added. The cwTrapIndex object contains the value of cwfFeederIfNum in the cwfFeederTable.

Table 7-9 Feeder-Related Traps (continued)

Name	Number	Specific-Varbinds	Description
cwFeederLMIDown	61004	cwTrapIndex	This trap is sent when the LMI for the feeder port is disconnected. The cwTrapIndex object contains the value of cwfFeederIfNum in the cwfFeederTable.
cwFeederConfigChange	61005	cwTrapIndex	This trap is sent when the feeder node configuration changes. This trap is sent when the following objects are modified in the feeder node: • cwfFeederName • cwfLanIP • cwfNetIP The object cwTrapIndex contains the value of cwfFeederIfNum in the cwfFeederTable.

PXM Traps and Alarms

The various PXM45 traps and alarms are listed in Table 7-10.

Table 7-10 PXM45 Traps and Alarms

Name	Number	Specific Varbinds	Description
cwChassisIntegrated Alarm	60001	shelfIntegratedAlarm	This trap is sent when the integrated alarm for the shelf is changed.
cwUserLogin	60002	userName	This trap is sent when a user logs in.
cwUserLogout	60003	userName	This trap is sent when a user logs out.
cwShelfRestart	60004	cefcModuleResetReason	This trap is sent when the shelf is rebooted.
cwNodeNameChange	60006	none	This trap is sent when the node name is modified.
cwIpAddressChange	60007	ipAdEntAddr, ifType	This trap is sent when the IP Address (Ethernet IP, ATM IP) is modified.
cwControllerAdd	60008	cwTrapIndex, cvcConfControllerType, cvcConfControllerLocation	This trap is sent when a VSI Controller is added.
cwControllerDelete	60009	cwTrapIndex, cvcConfControllerType, cvcConfControllerLocation	This trap is sent when a VSI Controller is deleted.
cwDiskDbIntegCheck Done	60024	none	This trap is sent when the integrity check is finished for all the configured modules in the shelf, and database corruption was not found on any slot. The entPhysicalParentRelPos object contains the physical slot number.

Table 7-10 PXM45 Traps and Alarms (continued)

Name	Number	Specific Varbinds	Description
cwDiskDbIntegCheck Failed	60025	`cwTrapOctetString`	This trap is sent when there is a corruption of configuration on disk. The trap shows the logical slots whose configuration stored on disk is corrupted. The `cwTrapOctetString` varbind is an octet string, which contains a list of slot numbers whose database is corrupted. This trap is sent immediately after the integrity check is finished checking `Disk Db` for all slots. The `entPhysicalParentRelPos` contains the physical slot number.
cwChassisTemperature Normal	60026	`cwTrapIndex`, `entSensorThresholdSeverity`	This trap is sent when the temperature is back to normal.
cwChassisTemperature AboveNormal	60027	`entSensorThresholdSeverity`, `cwTrapIndex`	This trap is sent when the temperature is above the threshold.
cwChassisDclevel Normal	60028	`entSensorThresholdSeverity`, `cwTrapIndex`	This trap is sent when the DC Level is back to normal.
cwChassisDclevelBelow Normal	60029	`entSensorThresholdSeverity`, `cwTrapIndex`	This trap is sent when the DC level is below the normal range.
cwChassisPowersupply Normal	60030	`entSensorThresholdSeverity`, `cwTrapIndex`	This trap is sent when the power supply is back to normal voltage.
cwChassisPowersupply Failed	60031	`entSensorThresholdSeverity`, `cwTrapIndex`	This trap is sent when the power supply has failed.
cwChassisFanrpm Normal	60032	`entSensorThresholdSeverity`, `cwTrapIndex`	This trap is sent when the fan RPM is back to normal.
cwChassisFanrpm BelowNormal	60033	`entSensorThresholdSeverity`, `cwTrapIndex`	This trap is sent when the fan RPM is below the normal value.
cwChassisDclevelAbove Normal	60034	`cwTrapIndex`	This trap is sent when the shelf DC level is above the normal DC level. The `cwTrapIndex` contains the `entPhysicalIndex` value for the DC level monitoring device.
cwModuleInserted	60051	`cwTrapIndex`	This trap is sent when the Front Module is plugged into a slot. It is also sent for the Processor Module as well as Service Modules when they are plugged in.
cwModuleRemoved	60052	`cwTrapIndex`	This trap is sent when the Front Module is removed from the slot. It is also sent for the Processor Module and service modules when they are removed.
cwModuleMismatch	60053	`cefcModuleOperStatus`, `cwTrapPhysicalVendorType`, `cwTrapIndex`	This trap is sent when the Front Module is plugged into a slot which was configured with another module earlier.

Table 7-10 PXM45 Traps and Alarms (continued)

Name	Number	Specific Varbinds	Description
cwModuleActive	60055	cwTrapIndex	This trap is sent when a module becomes active. This happens due to a switchover or when a failed card becomes active.
cwModuleStandby	60056	cwTrapIndex	This trap is sent when a module becomes standby due to switchover.
cwModuleFailed	60057	cefcModuleOperStatus, cwTrapIndex	This trap is sent when a module goes to a failed state.
cwLineModuleInserted	60058	cwTrapPhysicalVendorType, cwTrapLineModuleNumber, cwTrapIndex	This trap is sent when a line module (back card) is inserted.
cwLineModuleRemoved	60059	cwTrapPhysicalVendorType, cwTrapIndex	This trap is sent when a line module (back card) is removed.
cwLineModule Mismatch	60060	cefcModuleOperStatus, cwTrapPhysicalVendorType, cwTrapLineModuleNumber, cwTrapIndex	This trap is sent when a incompatible line module (back card) is inserted into the front module or an inserted line module is not the same as the one configured earlier.
cwLineModule MismatchClear	60062	cefcModuleOperStatus, cwTrapPhysicalVendorType, cwTrapLineModuleNumber, cwTrapIndex	This trap is sent when a line module mismatch is cleared. entPhysicalParentRelPos contains the physical slot number of the module. entPhysicalVendorType contains the type of the module. After a mismatch is cleared, cwTrapPhysicalVendorType contains the line module type. cwTrapLineModuleNumber contains the line module number. cwTrapIndex contains the entPhysicalIndex value for the line module that was in a mismatch, but cleared now.
cwCoreCardSwitch	60078	cwTrapSlotNumber, cwTrapIndex	This trap is sent when there is a switchover for core card set.
cwRedundancyAdd	60079	cwTrapSlotNumber	This trap is sent when redundancy is added for a module pair.
cwRedundancyDelete	60080	cwTrapSlotNumber	This trap is sent when a module pair is deleted from the redundancy configuration.
cwRedundancyActivate	60081	cwTrapSlotNumber	This trap is sent when a secondary module becomes active. This is due to a switchover.
cwRedundancyRevert	60082	cwTrapSlotNumber	This trap is sent when a primary module becomes active. This is due to a switchover.

Table 7-10 PXM45 Traps and Alarms (continued)

Name	Number	Specific Varbinds	Description
cwTrapsLost[1]	60900	none	This trap is sent when some traps could not be sent to the network management stations due to following reasons: • Trap could not be sent from the service module to the processor module. • Trap buffer is overflowed.
cwBulkFileCreation Started	60901	`cbfDefineMaxFiles`	This trap is sent when the bulk file creation is started.
cwBulkFileCreation Done	60902	`cbfDefineMaxFiles`, `cbfDefineFileName`, `cbfStatusFileCompletionTime`	This trap is sent when the bulk file creation is done.
cwBulkFileCreation Aborted	60903	`cbfDefineMaxFiles`, `cbfStatusFileState`	This trap is sent when the bulk file creation is aborted.
cwBulkNoTask Available	60904	`cbfDefineMaxFiles`	This trap is sent when the bulk file creation fails due to nonavailability of the task.
cwBulkFileCreation Failed	60905	`cbfDefineMaxFiles`	This trap is sent when the bulk file creation is fails for reasons other than the nonavailability of the task.
cwStatScmIpAddress Change	60920	`cwTrapIndex`, `cwsConfIpAddress`	This trap is sent for the following conditions: • SCM IP address is added. • SCM IP address is deleted. • SCM IP address is modified. `cwTrapIndex` contains the value of the `cwsConfIndex`, which contains the IP address entry added, deleted, or modified. When the IP address is added or modified, the `cwConfIpAddress` contains the value of the IP address. When the IP address is deleted, the `cwConfIpAddress` contains the value of zero for the IP address.
cwStatFileInfo	60921	`cwsStatsFileInfo`	This trap is sent for the following conditions: • One or more statistics files are created. • In a certain time, one or more statistics files are not uploaded by SCM since it was created.

1. This trap is also applicable to PXM1, AXSM, and AXSM-E.

PNNI Traps and Alarms

PNNI traps and alarms are listed in Table 7-11.

Note PNNI traps and alarms are supported by PXM45.

Table 7-11 PNNI List of Traps

Name	Number	Specific Varbinds	Description
ciscoWANSscopLink Change	70003	`ifIndex, cwspPhyIdentifier, cwspSscopStatus`	This trap is sent if the `Sscop` link status is failed or established.
ciscoWANControllerState Change	70004	`cwspPnniControllerStatus, cwspPnniControllerPhySlot`	This trap is sent when the PNNI controller is either active or standby.
ciscoWANIntfAddTrap	70005	`ifIndex, cwspPhyIdentifier`	This trap is sent to add an interface.
ciscoWANIntfDeleteTrap	70006	`ifIndex, cwspPhyIdentifier`	This trap is sent to delete an interface.
ciscoWANIntfConfigurtaionChangeTrap	70007	`ifIndex, cwspPhyIdentifier`	This trap is sent to change the configuration for an interface. The `ciscoWANIntfConfigurtaionChangeTrap` trap is received when a SVC port named `nni_version` is modified.
ciscoWANIntfOperation ChangeTrap	70008	`ifIndex, cwspPhyIdentifier, cwspOperStatus, cwspOperIfcType, cwspOperFailReason`	This trap is sent to change the operation status for an interface.
ciscoWANSpvcFailTrap	70009	`ifIndex, cwspConnTraceSrcVpi, cwspConnTraceSrcVci, cwspCallCallingAddress, ciscoWANSpvcFailReason`	This trap is sent when the Soft Permanent Virtual Connection (SPVC) failure is recognized in the PNNI controller.
ciscoWANPnniController StndbyFailed	70010	`cwspPnniControllerPhySlot`	This trap is sent when the PNNI standby controller fails.
ciscoWANSpvcNodePrefix Change	70011	`cwspSpvcNodePrefix`	This trap is sent when the SPVC node prefix is changed. The `cwspSpvcNodePrefix` can contain the updated prefix.

Table 7-11 PNNI List of Traps (continued)

Name	Number	Specific Varbinds	Description
ciscoWANSpvcFailure Trap	70012	`ifIndex, cwspPhyIdentifier, cwspConnTraceSrcVpi, cwspConnTraceSrcVci, cwspCallCallingAddress, ciscoWANSpvcFailReason`	This trap is sent when the PNNI controller sends this notification when the SPVC manager for the controller recognizes a failure. This trap deprecates trap ciscoWANSpvcFailTrap (70009). **Note** All the MIB objects here are for trap usage only. SNMP `Get` and `GetNext` is not meaningful for those objects.
ciscoWANClockChange Trap	70013	`cwTrapDisplayString`	This trap is sent when the PNNI controller sends this notification when the clock source changed on the switch. The `cwTrapDisplayString` varbind contains a description of the new clock source. This string contains the port number of the clock source or the string internal oscillator. The following are the port numbers: • 7.35 E1 • 7.35 T1 • 7.36 E1 • 7.36 T1 or in the format of slot:subslot.port:subport. For example, 5:1.2:2 or internal oscillator for the internal clock.

RPM Traps

The Router Processor Module (RPM) traps include the subinterface-related traps.

RPM Subinterface-Related Traps

The RPM subinterface-related traps are used to report any status changes on an RPM backplane subinterface. These traps are listed in Table 7-12.

Table 7-12 RPM Subinterface-Related Traps

Name	Number	Specific Varbinds	Description
cwRpmSubIfAdd	60401	cwTrapIndex, ifIndex, cwrSubIfAdapterIf	This trap is sent when a subinterface is added. The cwTrapIndex object contains the cwrSubIfSlotNum value. The ifIndex object contains the cwrSubIfNum value.
cwRpmSubIfDelete	60402	cwTrapIndex, ifIndex, cwrSubIfAdapterIf	This trap is sent when a subinterface is deleted. The cwTrapIndex object contains the cwrSubIfSlotNum value. The ifIndex object contains the cwrSubIfNum value.
cwRpmSubIfUp	60403	cwTrapIndex, ifIndex, cwrSubIfAdapterIf	This trap is sent when a subinterface is up. The cwTrapIndex object contains the cwrSubIfSlotNum value. The ifIndex object contains the cwrSubIfNum value.
cwRpmSubIfDown	60404	cwTrapIndex, ifIndex, cwrSubIfAdapterIf	This trap is sent when a subinterface is down. The cwTrapIndex object contains the cwrSubIfSlotNum value. The ifIndex object contains the cwrSubIfNum value.
cwRpmSubIfConfig Change	60405	cwTrapIndex, ifIndex, cwrSubIfAdapterIf	This trap is sent when a subinterface configuration is changed. The cwTrapIndex object contains the cwrSubIfSlotNum value. The ifIndex object contains the cwrSubIfNum value.
cwRpmSubIfOperUp	60406	cwTrapIndex, ifIndex, cwrSubIfAdapterIf	This trap is sent when a subinterface operational status is upped. The cwTrapIndex object contains the cwrSubIfSlotNum value. The ifIndex object contains the cwrSubIfNum value.
cwRpmSubIfOper Down	60407	cwTrapIndex, ifIndex, cwrSubIfAdapterIf	This trap is sent when a subinterface operational status is downed. The cwTrapIndex object contains the cwrSubIfSlotNum value. The ifIndex object contains the cwrSubIfNum value.
cwRpmSubIfOper Unknown	60408	cwTrapIndex, ifIndex, cwrSubIfAdapterIf	This trap is sent when a subinterface operational status is other than up or down. The cwTrapIndex object contains the cwrSubIfSlotNum value. The ifIndex object contains the cwrSubIfNum value.

Mandatory Trap Varbinds

Table 7-13 lists the trap varbinds in each trap generated by the switch. Each of these varbinds, in the order given, is available for each trap generated.

Table 7-13 Mandatory Trap Varbinds

Varbind	Data Type	Description
lastSequenceNumber	INTEGER	Contains the sequence number assigned to the trap. This can be used by NMS applications to implement a Robust Trap Mechanism.
sysName	OctetString	Contains the node name of the switch.
entPhysicalContainedIn	INTEGER(0..2147483647)	Contains the chassis number into which the module is connected.
entPhysicalParentRelPos	INTEGER(-1..2147483647)	Contains the physical slot number to into which the module is connected.
entPhysicalVendorType	ObjectID	Contains the type of module for which the trap is generated.
moduleTrapAlarm Severity	INTEGER (enumeration)	Contains the alarm severity of the trap.
genericTimeStamp	OctetString(1..30)	Contains the time when the trap is generated.

Trap-Specific Varbinds

The trap-specific varbinds used for the applicable service module are listed in Table 7-14.

Table 7-14 Trap-Specific Varbinds

Varbind	Syntax	Description	Service Module
ifIndex	Integer32(1..2147483647)	Specifies a unique value, greater than 0, for each interface. It is recommended that values are assigned starting with 1.	AXSM PXM45 RPM
ifName	OctetString(0..255)	Indicates the textual name of the interface.	AXSM PXM45
sonetSectionCurrentStatus	Integer32(1..6)	Indicates the section status of the interface.	AXSM
sonetLineCurrentStatus	Integer32(1..6)	Indicates the line status of the interface.	AXSM
sonetPathCurrentStatus	Integer32(1..62)	Indicates the path status of the interface	AXSM
dsx3LineStatus	INTEGER(1..4095)	Indicates the line status of the interface. It is used for loopback state information and failure state information.	AXSM
csApsProtectionIndex	Integer32(1..2147483647)	Indicates that the protection line becomes an active line when an APS switch occurs (APS switch can occur because of a failure on the working line).	AXSM
csApsActiveLine	INTEGER(enumeration)	Indicates which line is currently active.	AXSM

Table 7-14 Trap-Specific Varbinds (continued)

Varbind	Syntax	Description	Service Module
csApsLineFailureCode	INTEGER(enumeration)	Specifies the SONET APS line failure code, or the failure encountered by the APS line.	AXSM
cssTraceFailure	INTEGER(enumeration)	Specifies that the value of this object is set to true when the SONET section received trace does not match the cssTraceToExpect.	AXSM
cspTraceFailure	INTEGER(enumeration)	Sets the value of this object to true when the trace received by the SONET Path does not match the cspTraceToExpect.	AXSM
csApsLineSwitchReason	INTEGER(enumeration)	When the working line on one end fails, its other end is told to do an APS switch.	AXSM
cds3PlcpLineAlarmState	Gauge32	Specifies a bitmap of the DS3 Line Alarms. A value of 0 for this object indicates no alarms.	AXSM
cds3PlcpLineStatistical AlarmState	Gauge32	Specifies a bitmap of the DS3 PLCP Line Statistical Alarms.	AXSM
cds3LineStatisticalAlarm State	Gauge32(0..2147483647)	Specifies a bitmap of the DS3 Line Statistical Alarms.	AXSM
caviFileId	Gauge32	Determines the ID of the file that holds module-specific configuration parameters for this ATM Virtual Interface.	AXSM
cwmIngressSCTFileId	Gauge32(0..255)	Defines the file ID of the Service Class Template (SCT) file.	AXSM
cwsLineStatAlarmStatus	Gauge32	Indicates the status of the SONET line.	AXSM
cwsPathStatAlarmStatus	Gauge32	Indicates the status of the interface. The cwsPathStatAlarmStatus is a bitmap represented as a sum; therefore, it can represent multiple defects simultaneously.	AXSM
cwTrapIndex	Integer32(0..2147483647)	Contains the value that is indexed to any table.	AXSM PXM45 RPM
cwfLMIType	INTEGER { feeder(1), xLMI(2)}	Identifies the Local Management Interface (LMI) type used for the network to connect to the LMI port.	AXSM
cwRsrcPartController	Unsigned32 (1.. 255)	Specifies the controller identifier.	AXSM
cwaChanLocalVpi	Gauge32(0..4095)	Identifies the internal VPI assigned to a local endpoint by the switch.	AXSM
cwaChanLocalVci	Gauge32(0..65535)	Identifies the internal VCI assigned to a local endpoint by the switch.	AXSM
cwaChanVpcFlag	INTEGER (enumeration)	Identifies whether or not there is a VP/VC endpoint.	AXSM
cwaChanIdentifier	Gauge32	Represents a resource at the switch that has been assigned for this connection (identified by cwaChanVpi and cwaChanVci).	AXSM
cwaChanUploadCounter	Gauge32	Tracks the number of configuration changes that happen on a cwaChanIdentifier.	AXSM

Table 7-14 Trap-Specific Varbinds (continued)

Varbind	Syntax	Description	Service Module
cwaChanOperStatus	CiscoWanOperStatus	Reflects the operational status of an endpoint.	AXSM RPM
cwaChanAlarm State	INTEGER (enumeration)	Defines the alarms associated with an endpoint.	AXSM
cwaChanEgressXmtState	INTEGER	Specifies the state of the transmit portion of the endpoint in the egress direction.	AXSM
cwaChanEgressRcvState	INTEGER	Specifies the state of the receive portion of the endpoint in the egress direction.	AXSM
cwaChanIngressXmtState	INTEGER	Specifies the state of the transmit portion of the endpoint in the ingress direction.	AXSM
cwaChanIngressRcvState	INTEGER	Specifies the state of the receive portion of the endpoint in the ingress direction.	AXSM
shelfIntegratedAlarm	INTEGER	Displays the alarm status of the shelf.	PXM45
userName	OctetString(1..20)	Displays the username of the person logging in or logging out.	PXM45
cefcModuleResetReason	INTEGER	Identifies the reason for the last reset performed on the module.	PXM45
ipAdEntAddr	IpAddress	Specifies the IP address to which this entry addressing information pertains.	PXM45
ifType	INTEGER	Indicates the type of interface.	PXM45
cvcConfControllerType	INTEGER	Identifies the controller type.	PXM45
cvcConfController Location	Integer32(1..2147483647)	Identifies the location of the controller.	PXM45
entSensorThresholdSeverity	INTEGER	Indicates the severity of the sensor threshold.	PXM45
cefcModuleOperStatus	INTEGER	Shows the operational state of the module.	PXM45
cwTrapPhysicalVendorType	ObjectID	Contains the entPhysicalVendorType value for the physical entity.	PXM45
cwTrapLineModuleNumber	Integer32(1..4)	Contains the value for the entPhysicalParentRelPos object for line modules.	PXM45
cwTrapSlotNumber	Integer32(1..64)	Contains the slot number of the module	PXM45
cwTrapOctetString	OCTET STRING (SIZE (0..1024))	Contains the octet string value.	PXM45
cbfDefineMaxFiles	Gauge32	Determines the maximum number of file definitions that this system can hold in the cbfDefineFile table. A value of 0 indicates no configured limit. **Note** This object can be read-only on some systems because it is applicable only in traps. If the number is changed, it does not disturb existing entries.	PXM45

Table 7-14 Trap-Specific Varbinds (continued)

Varbind	Syntax	Description	Service Module
cbfDefineFileName	OctetString(1..255)	Creates the filename. Explicit device or path choices in the value of this object can override cbfDefineFileStorage.	PXM45
cwspPhyIdentifier	OctetString(0..255)	Indicates the physical identification of the interface.	PXM45
cwspSscopStatus	INTEGER	Specifies the Sscop link status on an Network-to-Network Interface (NNI).	PXM45
cwspPnniControllerStatus	INTEGER (enum)	Specifies the administrative status of the PNNI controller.	PXM45
cwspPnniControllerPhySlot	Integer32	Specifies the PNNI controller physical location.	PXM45
cwspOperStatus	INTEGER	Specifies the operational state of the interface.	PXM45
cwspOperIfcType	INTEGER	Specifies the interface type.	PXM45
cwspOperFailReason	INTEGER	Defines the SVC port failures. **Note** This object is used only for a trap varbind. NMS applications should not use issue SNMP Get and GetNextRequests for this object.	PXM45
cwspConnTraceSrcVpi	Integer32(0..4095)	Shows the VPI value of the starting point on this interface.	PXM45
cwspConnTraceSrcVci	Integer32(0..65535)	Shows the VCI value of the starting point on this interface.	PXM45
cwspCallCallingAddress	OctetString	Shows the calling party address of the call.	PXM45
ciscoWANSpvcFailReason	INTEGER (enumeration)	Defines the SPVC manager failures. **Note** This object is used only for a trap varbind. NMS applications should not depend on the implementation of this object. SNMP Get and GetNextRequests are not valid for this object.	PXM45
cwspSpvcNodePrefix	OctetString(13..13)	Specifies that write access is currently not permitted.	PXM45
cwrSubIfAdapterIf	Unsigned32	Maps to the RPM for port adapter interface.	RPM
cwTrapDisplayString	DisplayString	Contains the display string value.	PXM45
cwsConfIpAddress	IpAddress	Specifies the IP address of the SCM.	PXM45

Table 7-14 Trap-Specific Varbinds (continued)

Varbind	Syntax	Description	Service Module

cwsStatsFileInfo	OCTET STRING (SIZE(0..512))	Contains information on statistics related files.
		The statistics files are uploaded and downloaded using a file transfer mechanism FTP, TFTP, and so forth. The downloaded and uploaded files are available from a nonvolatile storage, for example, hard disk, flash disk, and so forth.
		The following categories are used for the statistics file:
		• `stats upload` file—Contains statistics data. These files can be uploaded only from the switch.
		• `stats enable` file—Contains the statistics that have to be enabled or disabled in one or more modules. These files can be downloaded as well as uploaded to or from the switch.
		This object can contain one or more records of files that were created, not uploaded in time, or uploaded and downloaded.
		Along with the filename, each record contains type of operation (created, uploaded, downloaded major trap, or critical trap) and number of applicable failed attempts.
		The values for each of the fields in the OCTET STRING depends upon the file categories `stats upload`, `stats enable`, and so forth.
		The following is the object layout:
		`NumOfRecords (SIZE(1))`
		`RECORD (SIZE(23))`
		`Filename STRING (SIZE(21))`
		`Reason INT (SIZE(1))`
		`NumOfFailUpload (SIZE(1))`
		The following are the definitions for the fields:
		• `NumOfRecords`—Designates the number of records. These records can all be the same type and can contain file uploaded information, or `stat enable` information, `stat file` trap information, or `stat enable` trap information.
		• `Reason`—Specifies the following fields:
		– 1—Indicates the file is created or uploaded.
		– 2—Indicates a MAJOR warning for the file not being uploaded.
		– 3—Indicates a CRITICAL warning for the file not being uploaded.
			PXM45

Table 7-14 Trap-Specific Varbinds (continued)

Varbind	Syntax	Description	Service Module
cwsStatsFileInfo (continued)		– 4—Indicates that the SNMP manager has downloaded the `stats enable` file. – 5—Indicates that the module, for example, Processor Module or Service Module, accepted the `stats enable` file without any error. The modules store the statistics information to enable or disable a dependent mechanism. This value cannot be set from the SNMP Manager. – 6—Indicates that the module did not accept the `stats enable` file due to some error. This value cannot be set from the SNMP manager. • `NumOfFailUpload`—Speechifies the value is set to 0 except or for the cases when the `Reason` field is either 1 (created), 2, 3 (`stat file` trap). The parameter can have a value greater than 0. Max-Access: read-write Status: current	

CHAPTER 8

Agent Capabilities Supported

This chapter describes the agent capabilities used for AXSM, AXSM-E, PXM45, and RPM service modules.

Contents of this chapter include:

- Overview
- Agent Capabilities MIBs

Overview

According to RFC 2580, the agent capabilities macro is a set of capabilities for a SNMP agent. Agent capabilities allow the SNMP agent to define the level of support needed for the applicable Management Information Base (MIB) group. The description is bound to the value of the `sysORID` object, which is the object identifier for the entry. For example, several MIB objects can have either restricted or augmented syntax or access levels. For more detailed information, refer to RFC 2580.

Agent Capabilities MIBs

The MIBs used for the agent capabilities contain information on the limitations to support the MIB objects. The agent capabilities used for the applicable service module are listed in Table 8-1.

Note The agent capabilities used for Release 2 are compatible for Release 2.1.

Table 8-1 MIBs Used for Agent Capabilities

MIB Name	Agent Capability MIB Name	Service Module
CISCO-MGX8800-TRAPS-MIB	CISCO-MGX8800-TRAPS-CAPABILITY.my	AXSM PXM45 RPM
ATM-MIB	CISCO-WAN-ATM-CAPABILITY.my	PXM45
CISCO-ATM-CONN-MIB	CISCO-WAN-AXSM-ATM-CONN-CAPABILITY.my	AXSM
CISCO-WAN-ATM-CONN-MIB	CISCO-AXSME-ATM-CONN-CAPABILITY.my	AXSM-E

Table 8-1 MIBs Used for Agent Capabilities (continued)

MIB Name	Agent Capability MIB Name	Service Module
CISCO-ATM-VIRTUAL-IF-MIB	CISCO-WAN-AXSM-ATM-VIRTUAL-IF-CAPABILITY.my	AXSM
	CISCO-AXSME-ATM-VIRTUAL-IF-CAPABILITY.my	AXSM-E
BASIS-SHELF-MIB	CISCO-WAN-BASIS-SHELF-CAPABILITY.my	PXM45
CISCO-BULK-FILE-MIB	CISCO-WAN-BULK-FILE-CAPABILITY.my	AXSM PXM45 RPM
CISCO-DS3-MIB	CISCO-WAN-CISCO-DS3-CAPABILITY.my	AXSM
	CISCO-AXSME-CISCO-DS3-CAPABILITY.my	AXSM-E
DS3-MIB	CISCO-WAN-DS3-CAPABILITY.my	AXSM
ENTITY-MIB	CISCO-WAN-ENTITY-CAPABILITY.my	PXM45
CISCO-ENTITY-FRU-CONTROL-MIB	CISCO-WAN-ENTITY-FRU-CAPABILITY.my	PXM45
IF-MIB	CISCO-WAN-IF-CAPABILITY.my[1]	AXSM
	CISCO-WAN-PXM-IF-CAPABILITY.my[2]	PXM45
	CISCO-AXSME-IF-CAPABILITY.my	AXSM-E
SNMPv2-MIB	CISCO-WAN-MGX8800-SNMPV2-CAPABILITY.my	PXM45
PNNI-MIB	CISCO-WAN-PNNI-CAPABILITY.my	PXM45
RS-232-MIB	CISCO-WAN-RS-232-CAPABILITY.my	PXM45 PXM1
CISCO-WAN-RSRC-PART-MIB	CISCO-WAN-RSRC-PART-CAPABILITY.my	AXSM
CISCO-WAN-RPM-RSRC-PART-MIB	CISCO-WAN-RPM-RSRC-PART-CAPABILITY.my	RPM
SONET-MIB	CISCO-WAN-SONET-CAPABILITY.my	AXSM
CISCO-WAN-SVC-MIB	CISCO-WAN-SVC-CAPABILITY.my	PXM45
CISCO-SYSTEM-MIB	CISCO-WAN-SYSTEM-CAPABILITY.my	PXM45
CISCO-WAN-RPM-ATM-CONN-MIB	CISCO-WAN-RPM-ATM-CONN-CAPABILITY.my	RPM
CISCO-SONET-MIB	CISCO-AXSM-CISCO-SONET-CAPABILITY.my	AXSM
	CISCO-AXSME-CISCO-SONET-CAPABILITY.my	AXSM-E
CISCO-WAN-MODULE-MIB	CISCO-AXSM-WAN-MODULE-CAPABILITY.my	AXSM
	CISCO-AXSME-WAN-MODULE-CAPABILITY.my	AXSM-E
CISCO-ATM-CELL-LAYER-MIB	CISCO-AXSME-ATM-CELL-LAYER-CAPABILITY.my	AXSM-E
CISCO-WAN-ATM-CONN-STAT-MIB	CISCO-AXSME-ATM-CONN-STAT-CAPABILITY.my	AXSM-E
CISCO-WAN-RPM-SUBIF-MIB	CISCO-WAN-RPM-SUBIF-CAPABILITY.my	RPM

1. The CISCO-WAN-IF-CAPABILITY.my file is used for AXSM.
2. The CISCO-WAN-PXM-IF-CAPABILiTY.my file is used for PXM45.

Cisco MGX 8800 Traps Capabilities

The following are the object groups applicable to the Cisco MGX8800 traps capabilities:

- cwSonetNotificationsGroup
- cwSonetStatsNotificationsGroup
- cwDs3NotificationsGroup
- cwDs3LoopbackNotificationsGroup
- cwDs3StatsNotificationsGroup
- cwDs3PlcpNotificationsGroup
- cwAtmIfNotificationsGroup
- cwCardIngNotificationsGroup
- cwRsrcPartNotificationsGroup
- cwVsiControllerNotificationsGroup
- cwFeederNotificationsGroup

The Cisco MGX 8800 Traps capabilities reside in the CISCO-MGX8800-TRAPS-CAPABILITY.my file. For more detailed description about each trap, see Chapter 7, "Traps and Alarms Supported."

The variation type of MIB objects are listed in Table 8-2.

Table 8-2 Cisco MGX8800 Traps Capabilities

Variation Name	Capability Specific Varbinds	MIB Specific Varbinds	MIB Access	Capability Access
cwAtmPhyInAlarm	none	ifIndex, ifName	none	not implemented
cwAtmPhyNoAlarm	none	ifIndex, ifName	none	not implemented

Cisco ATM Capabilities

The atmInterfaceConfGroup2 object group is applicable to Cisco WAN ATM Capabilities. The Cisco ATM capabilities reside in the CISCO-WAN-ATM-CAPABILITY.my file. For a more detailed description about each variation object, see Chapter 5, "PNNI MIB Objects".

The variation type of MIB objects are listed in Table 8-3.

Table 8-3 Cisco WAN ATM Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
atmInterfaceMaxVpcs	none	INTEGER (0..4096)	Specifies that the atmInterfaceMaxVpcs variation object is not supported. Max-Access: read-write Access: not implemented
atmInterfaceMaxVccs	none	INTEGER (0..65536)	Specifies that the atmInterfaceMaxVccs variation object is not supported. Max-Access: read-write Access: not implemented

Table 8-3 Cisco WAN ATM Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
atmInterfaceConfVpcs	none	`INTEGER (0..4096)`	Specifies that the `atmInterfaceConfVpcs` variation object is not supported. Max-Access: read-only Access: not implemented
atmInterfaceConfVccs	none	`INTEGER (0..65536)`	Specifies that the `atmInterfaceConfVccs` variation object is not supported. Max-Access: read-only Access: not implemented
atmInterfaceMaxActiveVpiBits	none	`INTEGER (0..12)`	Specifies that the `atmInterfaceMaxActiveVpiBits` variation object is not supported. Max-Access: read-write Access: not implemented
atmInterfaceMaxActiveVciBits	none	`INTEGER (0..16)`	Specifies that the `atmInterfaceMaxActiveVciBits` variation object is not supported. Max-Access: read-write Access: not implemented
atmInterfaceIlmiVpi	none	`AtmVpIdentifier`	Specifies that the `atmInterfaceIlmiVpi` variation object is not supported. Max-Access: read-write Access: not implemented
atmInterfaceIlmiVci	none	`AtmVcIdentifier`	Specifies that the `atmInterfaceIlmiVci` variation object is not supported. Max-Access: read-write Access: not implemented
atmInterfaceMyNeighborIpAddress	none	`IpAddress`	Specifies that write access is not supported. Max-Access: read-write Access: read-only
atmInterfaceMyNeighborIfName	none	`DisplayString`	Specifies that write access is not supported. Max-Access: read-write Access: read-only
atmInterfaceCurrentMaxVpiBits	none	`INTEGER (0..12)`	Specifies that the `atmInterfaceCurrentMaxVpiBits` variation object is not supported. Max-Access: read-only Access: not implemented

Table 8-3 Cisco WAN ATM Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
atmInterfaceCurrentMaxVciBits	none	`INTEGER (0..16)`	Specifies that the `atmInterfaceCurrentMaxVciBits` variation object is not supported. Max-Access: read-only Access: not implemented
atmInterfaceSubscrAddress	none	`AtmAddr`	Specifies that the `atmInterfaceSubscrAddress` variation object is not supported. Max-Access: read-write Access: not implemented

Cisco ATM Connections Capabilities

The following are the object groups applicable to Cisco WAN AXSM ATM CONN capabilities:

- `ciscoWanAtmConnChanMIBGroup`
- `ciscoWanAtmConnStateGroup`

The Cisco ATM Connections capabilities reside in the CISCO-WAN-AXSM-ATM-CONN-CAPABILITY.my file.

For a more detailed description about each variation object, see Chapter 3, "Cisco Enterprise MIB Objects."

The variation type of MIB objects are listed in Table 8-4.

Table 8-4 Cisco WAN AXSM ATM Connections Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
cwaChanControllerId	none	`Unsigned32 (1..255)`	Specifies that the value used for a write operation is ignored. The read operation returns the controller ID configured in the shelf. Max-Access: read-create Access: read-create
cwaChanPCR	`Unsigned32 (7..5651328)`	`Unsigned32 (0..4294967295)`	Specifies write support for a restricted range. Max-Access: read-create Access: read-create
cwaChanMCR	`Unsigned32 (7..5651328)`	`Unsigned32 (0..4294967295)`	Specifies write support for a restricted range. Max-Access: read-create Access: read-create

Table 8-4 Cisco WAN AXSM ATM Connections Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
cwaChanSCR	Unsigned32 (7..5651328)	Unsigned32 (0..4294967295)	Specifies write support for a restricted range. Max-Access: read-create Access: read-create
cwaChanRemotePCR	Unsigned32 (7..5651328)	Unsigned32 (0..4294967295)	Specifies write support for a restricted range. Max-Access: read-create Access: read-create
cwaChanRemoteMCR	Unsigned32 (7..5651328)	Unsigned32 (0..4294967295)	Specifies write support for a restricted range. Max-Access: read-create Access: read-create
cwaChanRemoteSCR	Unsigned32 (7..5651328)	Unsigned32 (0..4294967295)	Specifies write support for a restricted range. Max-Access: read-create Access: read-create
cwaChanAbrICR	Unsigned32 (7..5651328)	Unsigned32 (0..4294967295)	Specifies write support for a restricted range. Max-Access: read-create Access: read-create
cwaChanCDV	Unsigned32(0..16777215)	Unsigned32 (0..16777215)	Specifies write support for a restricted range. Max-Access: read-create
cwaChanCTD	Unsigned32(0..65535)	Unsigned32 (0..65535)	Specifies write support for a restricted range. Max-Access: read-create
cwaChanMBS	Unsigned32(0..5000000)	Unsigned32 (0..5000000)	Specifies write support for a restricted range. Max-Access: read-create
cwaChanCDVT	Unsigned32(0..5360000)	Unsigned32 (0..4294967295)	Specifies write support for a restricted range. Max-Access: read-create
cwaChanRemoteCDV	Unsigned32(0..16777215)	Unsigned32 (0..16777215)	Specifies write support for a restricted range. Max-Access: read-create
cwaChanRemoteCTD	Unsigned32(0..65535)	Unsigned32 (0..65535)	Specifies write support for a restricted range. Max-Access: read-create

Table 8-4 Cisco WAN AXSM ATM Connections Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
cwaChanRemoteMBS	`Unsigned32(0..5000000)`	`Unsigned32 (0..5000000)`	Specifies write support for a restricted range. Max-Access: read-create
cwaChanRemoteCDVT	`Unsigned32(0..5360000)`	`Unsigned32 (0..4294967295)`	Returns the value `0xFFFFFFFF` on a read operation. Max-Access: read-create
cwaChanPercentUtil	none	`Unsigned32 (0..100)`	Specifies the value in a write operation is ignored. The read operation returns the default value. Max-Access: read-create Access: read-create
cwaChanRemotePercent Util	none	`Unsigned32 (0..100)`	Specifies the value in a write operation is ignored. The read operation returns the default value. Max-Access: read-create Access: read-create
cwaChanRowStatus	`INTEGER {active(1), notInService(2), createAndGo(4), destroy(6)}`	`RowStatus`	Specifies the value `createAndWait(5)` is not supported. Max-Access: read-create
cwaChanIntAbrVSVD	none	`CiscoWanVSVDConfg`	Specifies the value in a write operation is ignored. The read operation returns the default value. Max-Access: read-create Access: read-create
cwaChanExtAbrVSVD	none	`CiscoWanVSVDConfg`	Specifies the value in a write operation is ignored. The read operation returns the default value. Max-Access: read-create Access: read-create
cwaChanAisIWCapability	none	`CiscoWanAisIW`	Specifies the value in a write operation is ignored. The read operation returns the default value. Max-Access: read-create Access: read-create
cwaChanCLR	none	`Unsigned32 (1..15)`	Specifies the value in a write operation is ignored. The read operation returns the default value. Max-Access: read-create Access: read-create

Table 8-4 Cisco WAN AXSM ATM Connections Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
cwaChanRemoteCLR	none	`Unsigned32 (1..15)`	Specifies the value in a write operation is ignored. The read operation returns the default value. Max-Access: read-create Access: read-create
cwaChanTestType	`INTEGER { noLpbk(1), nonDestructive(3) }`	`CiscoWanLpbkTypes`	Specifies the value `destructive(2)` is not supported. Max-Access: read-create
cwaChanTestIterations	`Unsigned32 (1..1)`	`Unsigned32 (1..255)`	Supports the value 1. Max-Access: read-create

Cisco AXSM-E ATM Connections Capabilities

This section describes the agent capabilities that support the Cisco WAN ATM Connections MIB for AXSM-E.

The following are the object groups applicable to the CISCO-AXSME-ATM-CONN-CAPABILITY.my file:

- `ciscoWanAtmConnChanMIBGroup`
- `ciscoWanAtmConnStateGroup`

For a more detailed description, see Chapter 3, "Cisco Enterprise MIB Objects."

The variation type of objects are listed in Table 8-5.

Table 8-5 Cisco AXSM-E ATM Connections Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
cwaChanControllerId	none	`Unsigned32 (1..255)`	Specifies that the value in the write operation is ignored. The read operation returns the controller ID configured in the shelf.
cwaChanPCR	`Unsigned32 (7..1412830)`	`Unsigned32 (0..4294967295)`	Specifies that the write operation is supported with the restricted range.
cwaChanMCR	`Unsigned32 (7..1412830)`	`Unsigned32 (0..4294967295)`	Specifies that the write operation is supported with the restricted range.
cwaChanSCR	`Unsigned32 (7..1412830)`	`Unsigned32 (0..4294967295)`	Specifies that the write operation is supported with the restricted range.
cwaChanRemotePCR	`Unsigned32 (7..1412830)`	`Unsigned32 (0..4294967295)`	Specifies that the write operation is supported with the restricted range.
cwaChanRemoteMCR	`Unsigned32 (7..1412830)`	`Unsigned32 (0..4294967295)`	Specifies that the write operation is supported with the restricted range.
cwaChanRemoteSCR	`Unsigned32 (7..1412830)`	`Unsigned32 (0..4294967295)`	Specifies that the write operation is supported with the restricted range.

Table 8-5 Cisco AXSM-E ATM Connections Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
cwaChanAbrICR	`Unsigned32 (7..1412830)`	`Unsigned32 (0..4294967295)`	Specifies that the write operation is supported with the restricted range.
cwaChanCDVT	`Unsigned32 (0..5360000)`	`Unsigned32 (0..4294967295)`	Specifies that the write operation is supported with the restricted range.
cwaChanRemote CDVT	`Unsigned32 (0..5360000)`	`Unsigned32 (0..4294967295)`	Specifies that the value 0xFFFFFFFF is always returned for the read operation.
cwaChanPercentUtil	none	`Unsigned32 (0..100)`	Specifies that the value in the write operation is ignored. The read operation returns the default value.
cwaChanRemote PercentUtil	none	`Unsigned32 (0..100)`	Specifies that the value in the write operation is ignored. The read operation returns the default value.
cwaChanRowStatus	`INTEGER { active(1), notInService(2), createAndGo(4), destroy(6) }`	`RowStatus`	Specifies that the `createAndWait(5)` value is not supported.
cwaChanAbrVSVD Enable	none	`TruthValue`	Specifies that the `cwaChanAbrVSVDEnable` variation object is not supported. Access: not-implemented
cwaChanAbrERS	none	`CiscoWanERSConfg`	Specifies that the value in the write operation is ignored. The read operation returns the default value.
cwaChanAisIW Capability	none	`CiscoWanAisIW`	Specifies that the value in the write operation is ignored. The read operation returns the default value.
cwaChanCLR	none	`Unsigned32 (1..15)`	Specifies that the value in the write operation is ignored. The read operation returns the default value.
cwaChanRemoteCLR	none	`Unsigned32 (1..15)`	Specifies that the value in the write operation is ignored. The read operation returns the default value.
cwaChanTestType	`INTEGER { noLpbk(1), nonDestructive(3) }`	`CiscoWanLpbkTypes`	Specifies that the value `destructive(2)` is not supported.
cwaChanTestIterations	`Unsigned32 (1..1)`	`Unsigned32 (1..255)`	Supports only value 1.

Cisco WAN RPM ATM Connection Capabilities

This section describes the agent capabilities that support the Cisco WAN ATM CONN MIB for Router Processor Module (RPM).

The following are the object groups applicable to the CISCO-WAN-RPM-ATM-CONN-CAPABILITY.my file:

- `ciscoWanAtmConnChanMIBGroup`
- `ciscoWanAtmConnStateGroup`

For a more detailed description, see Chapter 3, "Cisco Enterprise MIB Objects."

The variation type of MIB objects are listed in Table 8-6.

Table 8-6 Cisco WAN RPM ATM Connection Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
cwaChanVpi	`Unsigned32(0..255)`	`Unsigned32 (0..4095)`	Supports the value 0 for the VCC. Max-Access: not-accessible
cwaChanVci	`Unsigned32(1..3808)`	`Unsigned32 (0..65535)`	Specifies that the `cwaChanVci` variation object is used for the VCC. Max-Access: not-accessible
cwaChanService Category	`INTEGER { vbr3nRt(7), ubr1(8), abr(10) }`	`CiscoAtmServiceCategory`	Specifies that the values `cbr1(1)`, `vbr1RT(2)`, `vbr2RT(3)`, `vbr3RT(4)`, `vbr1nRT(5)`, `vbr2nRT(6)`, `ubr2(9)`, `cbr2(11)`, and `cbr3(12)` are not supported. Max-Access: read-create
cwaChanIdentifier	`Unsigned32(17..4095)`	`Unsigned32 (0..4294967295)`	Specifies that the supported range is 17 to 3824 for VCC, and the supported range is 3841 to 4095 for VPC. Max-Access: read-only
cwaChanStats Enable	none	`TruthValue`	Specifies that the read operation returns a value `false(2)`. Write operation is not supported. Max-Access: read-create
cwaChanLocalVpi	`Unsigned32(0..255)`	`Unsigned32 (0..4095)`	Specifies that the VPI value is set to zero for VCC. Max-Access: read-only
cwaChanLocalVci	`Unsigned32(1..3808)`	`Unsigned32 (0..65535)`	Determines the valid range for VCC. Max-Access: read-only
cwaChanRemote Vpi	`Unsigned32(0..255)`	`Unsigned32 (0..4095)`	Specifies that the VPI value is set to 0 for VCC. Max-Access: read-create
cwaChanRemote Vci	`Unsigned32(1..3808)`	`Unsigned32 (0..65535)`	Determines the valid range for VCC. Max-Access: read-create

Table 8-6 Cisco WAN RPM ATM Connection Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
cwaChanCCEnable	none	`TruthValue`	Specifies that the read operation returns a value `false(2)`. Write operation is not supported. Max-Access: read-create Access: read-only
cwaChanFrame Discard	none	`TruthValue`	Specifies that the read operation returns a value `false(2)`. This value is ignored for the write operation. Max-Access: read-create
cwaChanPCR	`Unsigned32(7..353208)`	`Unsigned32 (0..4294967295)`	Specifies that the `cwaChanPCR` variation object is not supported. Max-Access: read-create
cwaChanMCR	`Unsigned32(7..353208)`	`Unsigned32 (0..4294967295)`	Supports the limited range only. Max-Access: read-create
cwaChanSCR	`Unsigned32(7..353208)`	`Unsigned32 (0..4294967295)`	Supports the limited range only. Max-Access: read-create
cwaChanCDV	none	`Unsigned32 (0..16777215)`	Specifies that the `cwaChanCDV` variation object cannot be created or modified. Max-Access: read-create Access: read-only
cwaChanCTD	none	`Unsigned32 (0..65535)`	Specifies that the `cwaChanCTD` variation object cannot be created or modified. Max-Access: read-create Access: read-only
cwaChanMBS	`Unsigned32(1..65535)`	`Unsigned32 (0..5000000)`	Supports only the limited range. Max-Access: read-create
cwaChanCDVT	none	`Unsigned32 (0..4294967295)`	Specifies that the `cwaChanCDVT` variation object cannot be created or modified. Access: read-only
cwaChanRemote PCR	`Unsigned32(7..353208)`	`Unsigned32 (0..4294967295)`	Supports only the limited range. Max-Access: read-create
cwaChanRemote MCR	`Unsigned32(7..353208)`	`Unsigned32 (0..4294967295)`	Supports only the limited range. Max-Access: read-create
cwaChanRemote SCR	`Unsigned32(7..353208)`	`Unsigned32 (0..4294967295)`	Supports only the limited range. Max-Access: read-create

Table 8-6 Cisco WAN RPM ATM Connection Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
cwaChanRemote CDV	none	`Unsigned32 (0..16777215)`	Specifies that the `cwaChanRemoteCDV` variation object cannot be created or modified. Access: read-only
cwaChanRemote CTD	none	`Unsigned32 (0..65535)`	Specifies that the `cwaChanRemoteCTD` variation object cannot be created or modified. Access: read-only
cwaChanRemote CDVT	none	`Unsigned32 (0..4294967295)`	Specifies that the `cwaChanRemoteCDVT` variation object cannot be created or modified. Access: read-only
cwaChanAbrICR	none	`Unsigned32 (0..4294967295)`	Specifies that the `cwaChanAbrICR` variation object cannot be created or modified. Access: read-only
cwaChanAbrADTF	none	`Unsigned32 (1..1023)`	Specifies that the `cwaChanAbrADTF` variation object cannot be created or modified. Access: read-only
cwaChanAbrRIF	none	`AbrRateFactors`	Specifies that the `cwaChanAbrRIF` variation object cannot be created or modified. Access: read-only
cwaChanAbrNRM	none	`INTEGER {nrm2(1), nrm4(2), nrm8(3), nrm16(4), nrm32(5), nrm64(6), nrm128(7), nrm256(8)}`	Specifies that the `cwaChanAbrNRM` variation object cannot be created or modified. Access: read-only
cwaChanAbrTRM	none	`INTEGER {trm0point78125(1), trm1point5625(2), trm3point125(3), trm6point25(4), trm12point5(5), trm25(6), trm50(7), trm100(8)}`	Specifies that the `cwaChanAbrTRM` variation object cannot be created or modified. Access: read-only
cwaChanAbrCDF	none	`INTEGER {cdf0(1), cdfOneOver64(2), cdfOneOver32(3), cdfOneOver16(4), cdfOneOver8(5), cdfOneOver4(6), cdfOneOver2(7), cdfOne(8)}`	Specifies that the `cwaChanAbrCDF` variation object cannot be created or modified. Access: read-only

Table 8-6 Cisco WAN RPM ATM Connection Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
cwaChanAbrFRTT	none	`Unsigned32 (0..16700000)`	Specifies that the `cwaChanAbrFRTT` variation object cannot be created or modified. Access: read-only
cwaChanAbrTBE	none	`Unsigned32 (0..16777215)`	Specifies that the `cwaChanAbrTBE` variation object cannot be created or modified. Access: read-only
cwaChanAbrERS	none	`CiscoWanERSConfg`	Specifies that the write operation is ignored.
cwaChanAbrVSVD Enable	none	`TruthValue`	Specifies that the write operation is ignored.
cwaChanIntAbr VSVD	none	`CiscoWanVSVDConfg`	Specifies that the write operation is ignored.
cwaChanExtAbr VSVD	none	`CiscoWanVSVDConfg`	Specifies that the write operation is ignored.
cwaChanAisIW Capability	none	`CiscoWanAisIW`	Specifies that the `cwaChanAisIWCapability` variation object cannot be created or modified. Access: read-only
cwaChanCLR	none	`Unsigned32 (1..15)`	Specifies that the `cwaChanCLR` variation object cannot be created or modified. Access: read-only
cwaChanRemote CLR	none	`Unsigned32 (1..15)`	Specifies that the `cwaChanRemoteCLR` variation object cannot be created or modified. Access: read-only
cwaChanEgress XmtState	none	`CiscoWanXmtState`	Specifies that the `cwaChanEgressXmtState` variation object always returns a value zero.
cwaChanEgressRcv State	none	`CiscoWanRcvState`	Specifies that the `cwaChanEgressRcvState` variation object always returns a value zero.
cwaChanIngress XmtState	none	`CiscoWanXmtState`	Specifies that the `cwaChanIngressXmtState` variation object always returns a value zero.
cwaChanIngress RcvState	none	`CiscoWanRcvState`	Specifies that the `cwaChanIngressRcvState` variation object always returns a value zero.

Cisco ATM Virtual IF Capabilities

The following are the object groups applicable to the Cisco WAN AXSM ATM Virtual IF Capabilities for AXSM:

- caviMIBGroup
- caviEgressStatMIBGroup
- caviIngressStatMIBGroup

The Cisco ATM Virtual IF capabilities reside in the CISCO-WAN-AXSM-ATM-VIRTUAL-IF-CAPABILITY.my file.

For a more detailed description about each variation object, see Chapter 3, "Cisco Enterprise MIB Objects."

The variation type of MIB objects are listed in Table 8-7.

Table 8-7 Cisco WAN AXSM ATM Virtual IF Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
caviIfType	INTEGER { uni(1), nni(2) vnni(3)}	INTEGER {uni(1), nni(2), vnni(3), vuni(4)}	Specifies that value vuni(4) is not supported.
caviMinRate	Unsigned32 (50..5651320)	Unsigned32	Determines that the upper limit is the maximum rate for OC48 and equals to 5651320 cells per second.
caviMaxRate	Unsigned32 (50..5651320)	Unsigned32	Determines that the upper limit is the maximum rate for OC48 and equals to 5651320 cells per second.
caviEgrRMCells	none	Counter32	Specifies that the caviEgrRMCells variation object is not supported. Access: not implemented
caviEgrXmtEFCICells	none	Counter32	Specifies that the caviEgrXmtEFCICells variation object is not supported. Access: not implemented
caviEgrRcvEFCICells	none	Counter32	Specifies that the caviEgrRcvEFCICells variation object is not supported. Access: not implemented
caviEgrXmtOAMCells	none	Counter32	Specifies that the caviEgrXmtOAMCells variation object is not supported. Access: not implemented
caviHEgrXmtClp0Cells	none	Counter32	Specifies that the caviHEgrXmtClp0Cells variation object is not supported. Access: not implemented
caviHEgrXmtClp1Cells	none	Counter32	Specifies that the caviHEgrXmtClp1Cells variation object is not supported. Access: not implemented

Table 8-7 Cisco WAN AXSM ATM Virtual IF Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
caviIngRMCells	none	Counter32	Specifies that the caviIngRMCells variation object is not supported. Access: not implemented
caviIngXmtEFCICells	none	Counter32	Specifies that the caviIngXmtEFCICells variation object is not supported. Access: not implemented
caviIngRcvEFCICells	none	Counter32	Specifies that the caviIngRcvEFCICells variation object is not supported. Access: not implemented
caviIngRcvOAMCells	none	Counter32	Specifies that the caviIngRcvOAMCells variation object is not supported. Access: not implemented
caviHIngRcvClp0Cells	none	Counter64	Specifies that the caviHIngRcvClp0Cells variation object is not supported. Access: not implemented
caviHIngRcvClp1Cells	none	Counter64	Specifies that the caviHIngRcvClp1Cells variation object is not supported. Access: not implemented

Cisco ATM Virtual IF Capabilities for AXSM-E

The following are the object groups applicable to the Cisco ATM Virtual IF MIB capabilities for AXSM-E:

- caviMIBGroup
- caviEgressStatMIBGroup
- caviIngressStatMIBGroup
- caviEgressIntervalMIBGroup

The Cisco ATM Virtual IF capabilities reside in the CISCO-AXSME-ATM-VIRTUAL-IF-CAPABILITY.my file.

For a more detailed description about each variation object, see Chapter 3, "Cisco Enterprise MIB Objects."

The variation type of MIB objects are listed in Table 8-8.

Table 8-8 Cisco AXSM-E ATM Virtual IF Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
caviIfType	INTEGER { uni(1), nni(2) vnni(3)}	INTEGER {uni(1), nni(2), vnni(3), vuni(4)}	Specifies that value vuni(4) is not supported.
caviMinRate	Unsigned32 (50..1412830)	Unsigned32	Determines that the upper limit is the maximum rate for OC12 and equals to 1412830 cells per second.
caviMaxRate	Unsigned32 (50..1412830)	Unsigned32	Determines that the upper limit is the maximum rate for OC12 and equals to 1412830 cells per second.
caviEgrRMCells	none	Counter32	Specifies that the caviEgrRMCells variation object is not supported. Access: not implemented
caviEgrXmtEFCICells	none	Counter32	Specifies that the caviEgrXmtEFCICells variation object is not supported. Access: not implemented
caviEgrRcvEFCICells	none	Counter32	Specifies that the caviEgrRcvEFCICells variation object is not supported. Access: not implemented
caviEgrXmtOAMCells	none	Counter32	Specifies that the caviEgrXmtOAMCells variation object is not supported. Access: not implemented
caviHEgrXmtClp0Cells	none	Counter32	Specifies that the caviHEgrXmtClp0Cells variation object is not supported. Access: not implemented
caviHEgrXmtClp1Cells	none	Counter32	Specifies that the caviHEgrXmtClp1Cells variation object is not supported. Access: not implemented
caviIngRMCells	none	Counter32	Specifies that the caviIngRMCells variation object is not supported. Access: not implemented
caviIngXmtEFCICells	none	Counter32	Specifies that the caviIngXmtEFCICells variation object is not supported. Access: not implemented
caviIngRcvEFCICells	none	Counter32	Specifies that the caviIngRcvEFCICells variation object is not supported. Access: not implemented
caviIngRcvOAMCells	none	Counter32	Specifies that the caviIngRcvOAMCells variation object is not supported. Access: not implemented

Table 8-8 Cisco AXSM-E ATM Virtual IF Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
caviHIngRcvClp0Cells	none	`Counter64`	Specifies that the `caviHIngRcvClp0Cells` variation object is not supported. Access: not implemented
caviHIngRcvClp1Cells	none	`Counter64`	Specifies that the `caviHIngRcvClp1Cells` variation object is not supported. Access: not implemented

Basis Shelf Capabilities

The following are the object groups applicable to Cisco WAN Basis Shelf Capabilities:

- `shelfInfoGroup`
- `shelfNodeGroup`
- `shelfStatsGroup`
- `shelfApsInfoGroup`
- `shelfAxisInfoGroup`
- `shelfClkRateGroup`
- `shelfRedundancyGroup`

For a more detailed description about each variation object, refer to the CISCO-WAN-BASIS-SHELF-CAPABILITY.my file.

The variation type of MIB objects are listed in Table 8-9.

Table 8-9 Cisco WAN Basis Shelf Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
shelfNumOfValidEntries	none	`INTEGER (0..64)`	Specifies the `shelfNumOfValidEntres` variation object is not supported. Access: not implemented
shelfDate	none	`DisplayString (SIZE (10))`	Specifies the `shelfDate` variation object is not supported. Access: not implemented
shelfTime	none	`DisplayString (SIZE (8))`	Specifies the `shelfTime` variation object is not supported. Access: not implemented

shelfTmZn	none	`INTEGER{`
		`gmt (1),`
		`est (2),`
		`cst (3),`
		`mst (4),`
		`pst (5),`
		`edt (6),`
		`cdt (7),`
		`mdt (8),`
		`pdt (9)}`
			Specifies that the `shelfTmZn` variation object is not supported.
			Access: not implemented

Table 8-9 Cisco WAN Basis Shelf Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
shelfTmZnGMTOff	none	`INTEGER (-12..12)`	Specifies that the `shelfTmZnGMTOff` variation object is not supported. Access: not implemented
shelfBkPlnType	none	`INTEGER (1..10)`	Specifies that the `shelfBkPlnType` variation object is not supported. Access: not implemented
shelfBkplnSerialNum	none	`DisplayString (SIZE (0..11))`	Specifies that the `shelfBkplnSerialNum` variation object is not supported. MIB Access: read-only Access: not implemented
shelfAlarmCardBitMap	none	`INTEGER`	Specifies that the `shelfAlarmCardBitMap` variation object is not supported. Access: not implemented
shelfSlotNum	none	`INTEGER (1..33)`	Specifies that the `shelfSlotNum` variation object is not supported. MIB Access: read-only Access: not implemented
shelfBkplnSerialNumDeprecated	none	`INTEGER (1..4)`	Specifies that the `shelfBkplnSerialNumDeprecated` variation object is not supported. MIB Access: read-only Access: not implemented

shelfFunctionModuleState	none	`INTEGER{`
		`nocard (1),`
		`standby (2),`
		`active (3),`
		`failed (4),`
		`selfTest (5),`
		`heldInReset (6),`
		`boot (7),`
		`mismatch (8),`
		`unknown (9),`
		`coreCardMismatch (10),`
		`blocked (11),`
		`reserved (12),`
		`-- hold state applies to PXM during Graceful Upgrade.`
		`hold (13)}`
			Specifies that the `shelfFunctionModuleState` variation object is not supported.
			Access: not implemented

Table 8-9 Cisco WAN Basis Shelf Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
shelfFunctionModuleType	none	`INTEGER {other (1), asc (2), pxm1 (1000), pxm1-2t3e3 (1001), pxm1-4oc3 (1002), pxm1-oc12 (1003), rpm (2000)} rpm-pr (2001)}`	Specifies that the `shelfFunctionModuleType` variation object is not supported. MIB Access: read-only Access: not implemented
shelfFunctionModuleHoldReset	none	`INTEGER { doNotHold (1), holdInReset(2) }`	Specifies that the `shelfFunctionModuleHoldReset` variation object is not supported. MIB Access: read-write Access: not implemented
redPrimaryType	none	`INTEGER {other (1), bsc (2), -- reserved for MGX8850 after 1000 pxm1 (1000), pxm1-2t3e3 (1001), pxm1-4oc3 (1002), pxm1-oc12 (1003), rpm (2000)} rpm-pr (2001)}`	Specifies that the `redPrimaryType` variation object is not supported. **Note** The `redPrimaryType` object is not supported by PXM45 for Release 2.1. MIB Access: read-only Access: not implemented
redSecondaryType	none	`INTEGER {other (1), bsc (2), -- reserved for MGX8850 after 1000 pxm1 (1000), pxm1-2t3e3 (1001), pxm1-4oc3 (1002), pxm1-oc12 (1003), rpm (2000)} rpm-pr (2001)}`	Specifies that the `redSecondaryType` variation object is not supported. **Note** The `redSecondaryType` object is not supported by PXM45 for Release 2.1. MIB Access: read-only Access: not implemented

Table 8-9 Cisco WAN Basis Shelf Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
redFeature	none	`INTEGER`	Specifies that the `redFeature` variation object is not supported. MIB Access: read-only Access: not implemented
redLineModuleType	none	`INTEGER { lm-DB15-4T1 (16), lm-DB15-4E1 (17), lm-BNC-4E1 (18), lm-DB15-4T1-R(19), lm-DB15-4E1-R(20), lm-BNC-4E1-R (21), lm-RJ48-8T1 (22), lm-RJ48-8E1 (23), lm-SMB-8E1 (24), lm-RJ48-T3T1 (25), lm-RJ48-E3E1 (26), lm-RJ48-T3E1 (27), lm-SMB-E3E1 (28), lm-RJ48-E3T1 (29), lm-SMB-T3E1 (30), lm-T3E3-D (32), lm-T3E3-B (33), lm-RJ48-8T1-R(48), lm-RJ48-8E1-R(49), lm-SMB-8E1-R (50), -- HSSI/X.21 Added by Suheel lm-HS1-4X21 (60), lm-HS1-3HSSI (61), -- HSSI/X.21 lm-HS1-4V35 (62)}`	Specifies that the `redLineModuleType` variation object is not supported. **Note** The `redLineModuleType` object is not supported by PXM45 for Release 2.1. MIB Access: read-only Access: not implemented

Cisco Bulk File Capabilities

The following are the object groups applicable to Cisco WAN Bulk File capabilities:

- ciscoBulkFileDefineGroup
- ciscoBulkFileStatusGroup

The Cisco Bulk File capabilities reside in the CISCO-WAN-BULK-FILE-CAPABILITY.my file.

For a more detailed description, refer to the CISCO-BULK-FILE-MIB.my file.

The variation type of MIB objects are listed in Table 8-10.

Table 8-10 Cisco WAN Bulk File Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
cbfDefineMaxFiles	none	Unsigned32 (0..4294967295)	Specifies that the cbfDefineMaxFiles variation object is not supported. Max-Access: read-write Access: not implemented
cbfDefineFiles	none	Gauge32	Specifies that the cbfDefineFiles variation object is not supported. Max-Access: read-only Access: not implemented
cbfDefineHighFiles	none	Gauge32	Specifies that the cbfDefineHighFiles variation object is not supported. Max-Access: read-only Access: not implemented
cbfDefineFilesRefused	none	Counter32	Specifies that the cbfDefineFilesRefused variation object is not supported. Max-Access: read-only Access: not implemented
cbfDefineMaxObjects	none	Unsigned32 (0..4294967295)	Specifies that the cbfDefineMaxObjects variation object is not supported. Max-Access: read-write Access: not implemented
cbfDefineObjects	none	Gauge32	Specifies that the cbfDefineObjects variation object is not supported. Max-Access: read-only Access: not implemented
cbfDefineHighObjects	none	Gauge32	Specifies that the cbfDefineHighObjects variation object is not supported. Max-Access: read-only Access: not implemented

Table 8-10 Cisco WAN Bulk File Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
cbfDefineObjectsRefused	none	`Counter32`	Specifies that the `cbfDefineObjectsRefused` variation object is not supported. Max-Access: read-only Access: not implemented
cbfDefineObjectClass	none	`INTEGER {object(1), lexicalTable(2), leastCpuTable(3)}`	Specifies that the `cbfDefineObjectClass` variation object is not supported. Max-Access: read-create Access: not implemented
cbfDefineObjectID	none	`OBJECT IDENTIFIER`	Specifies that the `cbfDefineObjectID` variation object is not supported. Max-Access: read-create Access: not implemented
cbfDefineObjectEntryStatus	none	`RowStatus`	Specifies that the `cbfDefineObjectEntryStatus` variation object is not supported. Max-Access: read-create Access: not implemented
cbfStatusMaxFiles	none	`Unsigned32 (0..4294967295)`	Specifies that the `cbfStatusMaxFiles` variation object is not supported. Max-Access: read-write Access: not implemented
cbfStatusFiles	none	`Gauge32`	Specifies that the `cbfStatusFiles` variation object is not supported. Max-Access: read-only Access: not implemented
cbfStatusHighFiles	none	`Gauge32`	Specifies that the `cbfStatusHighFiles` variation object is not supported. Max-Access: read-only Access: not implemented
cbfStatusFilesBumped	none	`Counter32`	Specifies that the `cbfStatusFilesBumped` variation object is not supported. Max-Access: read-only Access: not implemented

Cisco DS3 Capabilities

The following are the object groups applicable to Cisco WAN Cisco DS3 Capabilities:

- ciscoDs3ConfMIBGroup
- cds3StatsMIBGroup
- ciscoDs3AlarmMIBGroup
- ciscoPlcpAlarmMIBGroup

The Cisco DS2 capabilities reside in the CISCO-WAN-CISCO-DS3-CAPABILITY.my file.

For a more detailed description, see Chapter 3, "Cisco Enterprise MIB Objects."

The variation type of MIB objects are listed in Table 8-11.

Table 8-11 Cisco WAN Cisco DS3 Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
cds3LineType	INTEGER {ds3cbitadm(1), ds3cbitplcp(2), e3g832adm(3) }	INTEGER {ds3cbitadm(1), ds3cbitplcp(2), e3g832adm(3)}	Specifies that the values e3g751adm(4), e3751plcp(5), ds3m23adm(6), ds3m23plcp(7), and other(8) are not supported. Max-Access: not-accessible
cds3InternalEqualizer	none	INTEGER {use(1), byPass(2)}	Specifies that the cds3InternalEqualizer variation object is not supported. Max-Access: read-write Access: not implemented
cds3NearEndLineLoopbackStatus	none	none	Specifies that the cds3NearEndLineLoopbackStatus variation object is not supported. Access: not implemented
cds3FarEndLineLoopbackStatus	none	none	Specifies that the cds3FarEndLineLoopbackStatus variation object is not supported. Access: not implemented

DS3 Capabilities

The following are the object groups applicable to Cisco WAN DS3 capabilities:

- ds3NearEndConfigGroup
- ds3NearEndStatisticsGroup

The DS3 capabilities reside in the CISCO-WAN-DS3-CAPABILITY.my file.

For a more detailed description about the variation objects, see Chapter 2, "Standard MIB Objects."

The variation type of MIB objects are listed in Table 8-12.

Table 8-12 Cisco WAN DS3 Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
dsx3LineType	none	INTEGER {dsx3other(1), dsx3M23(2), dsx3SYNTRAN(3), dsx3CbitParity(4), dsx3ClearChannel(5), e3other(6), e3Framed(7), e3Plcp(8)}	Specifies that write access is not supported. Max-Access: read-write Access: read-only
dsx3Channelization	none	INTEGER {disabled(1), enabledDs1(2), enabledDs2(3)}	Specifies that write access is not supported. Max-Access: read-write Access: read-only
dsx3LoopbackConfig	INTEGER { dsx3NoLoop(1), dsx3LineLoop(3), dsx3InwardLoop(5)}	INTEGER {dsx3NoLoop(1), dsx3PayloadLoop(2), dsx3LineLoop(3), dsx3OtherLoop(4), dsx3InwardLoop(5), dsx3DualLoop(6)}	Specifies that the values dsx3PayloadLoop(2), dsx3OtherLoop(4), and dsx3DualLoop(6) are not supported. Max-Access: read-write
dsx3Ds1ForRemoteLoop	none	INTEGER (0..29)	Specifies that the dsx3Ds1ForRemoteLoop variation object is not supported. Max-Access: read-write Access: not implemented
dsx3TransmitClockSource	INTEGER { loopTiming(1), localTiming(2) }	INTEGER {loopTiming(1), localTiming(2), throughTiming(3)}	Specifies that the value throughTiming(3) is not supported. Max-Access: read-write

Cisco DS3 Capabilities for AXSM-E

The following are the object groups applicable to Cisco AXSM-E Cisco DS3 capabilities:

- ciscoDs3ConfMIBGroup
- cds3StatsMIBGroup
- ciscoDs3AlarmMIBGroup
- ciscoPlcpAlarmMIBGroup
- ciscoPlcpCounterMIBGroup
- ciscoDs3Previous24HrGroup

The DS3 capabilities for AXSM-E reside in the CISCO-AXSME-CISCO-DS3-CAPABILITY.my file.

For a more detailed description about the variation objects, see Chapter 2, “Standard MIB Objects.”

The variation type of MIB objects are listed in Table 8-13.

Table 8-13 Cisco AXSM-E DS3 Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
cds3LineType	`INTEGER { ds3cbitadm(1), ds3cbitplcp(2), e3g832adm(3) }`	`INTEGER {ds3cbitadm(1), ds3cbitplcp(2), e3g832adm(3)}`	Specifies that the values `e3g751adm(4)`,`e3751plcp(5)`, `ds3m23adm(6)`,`ds3m23plcp(7)`, and `other(8)` are not supported.
cds3InternalEqualizer	none	`INTEGER { use(1), byPass(2) }`	Specifies that the `cds3InternalEqualizer` variation object is not supported. Access: not-implemented
cds3NearEndLineLoopbackStatus	none	`Unsigned32 (0..2147483647)`	Specifies that the `cds3NearEndLineLoopbackStatus` variation object is not supported. Access: not-implemented
cds3NEAlarmUpCount	none	`Unsigned32 (0..2147483647)`	Specifies that the `cds3NEAlarmUpCount` variation object is not supported. Access: not-implemented
cds3NEAlarmDownCount	none	`Unsigned32 (0..2147483647)`	Specifies that the `cds3NEAlarmDownCount` variation object is not supported. Access: not-implemented
cds3NEAlarmThreshold	none	`Unsigned32 (0..2147483647)`	Specifies that the `cds3NEAlarmThreshold` variation object is not supported. Access: not-implemented
cds3FEAlarmUpCount	none	`Unsigned32 (0..2147483647)`	Specifies that the `cds3FEAlarmUpCount` variation object is not supported. Access: not-implemented
cds3FEAlarmDownCount	none	`Unsigned32 (0..2147483647)`	Specifies that the `cds3FEAlarmDownCount` variation object is not supported. Access: not-implemented
cds3FEAlarmThreshold	none	`Unsigned32 (0..2147483647)`	Specifies that the `cds3FEAlarmThreshold` variation object is not supported. Access: not-implemented
cds3FECount	none	`Counter32`	Specifies that the `cds3FECount` variation object is not supported. Access: not-implemented

Table 8-13 Cisco AXSM-E DS3 Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
cds3EXZSCount	none	`Counter32`	Specifies that the `cds3EXZSCount` variation object is not supported. Access: not-implemented
cds3LCVCount	none	`Counter32`	Specifies that the `cds3LCVCount` variation object is not supported. Access: not-implemented
cds3PCVCount	none	`Counter32`	Specifies that the `cds3PCVCount` variation object is not supported. Access: not-implemented
cds3CPECount	none	`Counter32`	Specifies that the `cds3CPECount` variation object is not supported. Access: not-implemented
cds3FEBECount	none	`Counter32`	Specifies that the `cds3FEBECount` variation object is not supported. Access: not-implemented
cds3RcvAISCount	none	`Counter32`	Specifies that the `cds3RcvAISCount` variation object is not supported. Access: not-implemented

Entity Capabilities

The following object groups are applicable to Cisco WAN Entity capability:

- `entityPhysicalGroup`
- `entityPhysical2Group`

The entity capabilities reside in the CISCO-WAN-ENTITY-CAPABILITY.my file.

For a more detailed description, see Chapter 4, “PXM MIB Objects.”

The variation type of MIB objects are listed in Table 8-14.

Table 8-14 Cisco WAN Entity Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
entPhysicalName	none	`SnmpAdminString`	Specifies that the `entPhysicalName` object is set to one of the following: • `entPhysicalClass`—Equal to `module(9)`. – Determines the slot number for the front module. – Determines the slot number as seen in rear view of the chassis for the line module, for example, back cards. • `entPhysicalClass`—Equal to `chassis(1)`. – Determines the chassis number, for example, shelf number. Max-Access: read-only
entPhysicalMfgName	none	`SnmpAdminString`	Sets the PCB part number at the 800 level if applicable. Max-Access: read-only
entPhysicalModelName	none	`SnmpAdminString`	Contains the zero-length string. Max-Access: read-only
entPhysicalAssetID	none	`SnmpAdminString (SIZE (0..32))`	Specifies that the `entPhysicalAssetID` variation object is not supported. Max-Access: read-write Access: not implemented

Cisco Entity FRU Control Capabilities

The `cefcMIBModuleGroup` object group is applicable to Cisco WAN Entity FRU capabilities, which reside in the CISCO-WAN-ENTITY-FRU-CAPABILITY.my file.

For a more detailed description, see Chapter 4, "PXM MIB Objects."

One variation type of the object is listed in Table 8-15.

Table 8-15 Cisco WAN Entity FRU Control Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
cefcModuleAdminStatus	none	`ModuleAdminType`	Specifies that write access is not supported. Max-Access: read-write Access: read-only

IF Capabilities

The agent capabilities for the IF-MIB contain three different files. They include:

- Cisco WAN PXM IF Capabilities used for PXM45
- Cisco WAN IF Capabilities used for AXSM
- Cisco AXSM-E IF Capabilities used for AXSM-E

Cisco WAN PXM IF Capabilities

The following are the object groups applicable to Cisco WAN PXM IF capabilities:

- `ifGeneralInformationGroup`
- `ifPacketGroup`

The Cisco WAN PXM IF capabilities reside in the CISCO-WAN-PXM-IF-CAPABILITY.my file.

For a more detailed description, see Chapter 2, "Standard MIB Objects."

The variation type of MIB objects are listed in Table 8-16.

Table 8-16 Cisco WAN PXM IF Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
ifTableLastChange	none	`TimeTicks`	Specifies that the `ifTableLastChange` variation object is not supported. Max-Access: read-only Access: not implemented
ifAdminStatus	none	`Integer {up (1), ready to pass packets down; (2), testing; (3) in some test mode}`	Specifies that the `ifAdminStatus` variation object cannot be written for the RS-232 interface, which is identified by the `ifType` value `rs232(33)`. Max-Access: read-write Access: read-write
ifOperStatus	`INTEGER { up(1), down(2), testing(3) }`	`Integer {up (1), ready to pass packets; down(2); testing (3), in some test mode; unknown (4), status can not be determined; dormant (5); notPresent (6), some component is missing; lowerLayerDown (7), down due to state of lower-layer interface(s)}`	Specifies that the `ifOperStatus` variation object is unable to detect all the states. Max-Access: read-only
ifLastChange	none	`TimeTicks`	Specifies that the value cannot reflect the change in the operational status of the interface. Max-Access: read-only Access: read-only

Table 8-16 Cisco WAN PXM IF Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
ifLinkUpDownTrap Enable	none	`Integer {enabled (1), disabled (2)}`	Specifies that the write operation is not supported. The default value is `disabled`. Max-Access: read-write Access: read-only
ifAlias	none	`DisplayString (SIZE(0..64))`	Specifies that the `ifAlias` variation object is not supported. Max-Access: read-write Access: not implemented
ifInOctets	none	`Counter32`	Supports the following `ifType` interfaces: • Ethernet interface—Determines the value `ethernetCsmacd(6)`. • Loopback interface—Determines the value `softwareLoopback(24)`. • SLIP Interface—Determines the value `slip(28)`. Max-Access: read-only Access: read-only
ifOutOctets	none	`Counter32`	Supports the following `ifType` interfaces: • Ethernet interface—Determines the value `ethernetCsmacd(6)`. • Loopback interface—Determines the value `softwareLoopback(24)`. • SLIP Interface—Determines the value `slip(28)`. Max-Access: read-only Access: read-only
ifInUnknownProtos	none	`Counter32`	Supports the following `ifType` interfaces: • Ethernet interface—Determines the value `ethernetCsmacd(6)`. • Loopback interface—Determines the value `softwareLoopback(24)`. • SLIP Interface—Determines the value `slip(28)`. Max-Access: read-only Access: read-only

Table 8-16 Cisco WAN PXM IF Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
ifInErrors	none	`Counter32`	Supports the following `ifType` interfaces: • Ethernet interface—Determines the value `ethernetCsmacd(6)`. • Loopback interface—Determines the value `softwareLoopback(24)`. • SLIP Interface—Determines the value `slip(28)`. Max-Access: read-only Access: read-only
ifOutErrors	none	`Counter32`	Supports the following `ifType` interfaces: • Ethernet interface—Determines the value `ethernetCsmacd(6)`. • Loopback interface—Determines the value `softwareLoopback(24)`. • SLIP Interface—Determines the value `slip(28)`. Max-Access: read-only Access: read-only
ifInUcastPkts	none	`Counter32`	Supports the following `ifType` interfaces: • Ethernet interface—Determines the value `ethernetCsmacd(6)`. • Loopback interface—Determines the value `softwareLoopback(24)`. • SLIP Interface—Determines the value `slip(28)`. Max-Access: read-only Access: read-only
ifInMulticastPkts	none	`Counter32`	Specifies that the `ifInMulticastPkts` variation object is not supported. Max-Access: read-only Access: not implemented
ifInBroadcastPkts	none	`Counter32`	Specifies that the `ifInBroadcastPkts` variation object is not supported. Max-Access: read-only Access: not implemented

Table 8-16 Cisco WAN PXM IF Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
ifInDiscards	none	`Counter32`	Supports the following `ifType` interfaces: • Ethernet interface—Determines the value `ethernetCsmacd(6)`. • Loopback interface—Determines the value `softwareLoopback(24)`. • SLIP Interface—Determines the value `slip(28)`. Max-Access: read-only Access: read-only
ifOutUcastPkts	none	`Counter32`	Supports the following `ifType` interfaces: • Ethernet interface—Determines the value `ethernetCsmacd(6)`. • Loopback interface—Determines the value `softwareLoopback(24)`. • SLIP Interface—Determines the value `slip(28)`. Max-Access: read-only Access: read-only
ifOutMulticastPkts	none	`Counter32`	Specifies that the `ifOutMulticastPkts` variation object is not supported. Max-Access: read-only Access: not implemented
ifOutBroadcastPkts	none	`Counter32`	Specifies that the `ifOutBroadcastPkts` variation object is not supported. Max-Access: read-only Access: not implemented
ifOutDiscards	none	`Counter32`	Supports the following `ifType` interfaces: • Ethernet interface—Determines the value `ethernetCsmacd(6)`. • Loopback interface—Determines the value `softwareLoopback(24)`. • SLIP Interface—Determines the value `slip(28)`. Max-Access: read-only Access: read-only
ifPromiscuousMode	none	`TruthValue`	Specifies that the write operation is not supported. Max-Access: read-write Access: read-only

Table 8-16 Cisco WAN PXM IF Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
linkUp	none	{ifIndex, ifAdminStatus, ifOperStatus}	Specifies that the linkUp trap is not supported. Status: current Access: not implemented
linkDown	none	{ifIndex, ifAdminStatus, ifOperStatus}	Specifies that the linkDown trap is not supported. Status: current Access: not implemented

Cisco WAN IF Capabilities Used for AXSM

The ifGeneralInformationGroup is the object group applicable to Cisco WAN IF capabilities used for AXSM. These capabilities reside in the CISCO-WAN-IF-CAPABILITY.my file.

For a more detailed description, see Chapter 2, "Standard MIB Objects."

The variation type of MIB objects are listed in Table 8-17.

Table 8-17 Cisco WAN IF Capabilities Used for AXSM

Variation Name	Capability Syntax	MIB Syntax	Description
ifTableLastChange	none	TimeTicks	Specifies that the ifTableLastChange variation object is not supported. Max-Access: read-only Access: not implemented
ifPhysAddress	none	PhysAddress	Specifies that the ifPhysAddress variation object is not supported. Max-Access: read-only Access: not implemented
ifAdminStatus	INTEGER { up(1), down(2) }	Integer {up (1), ready to pass packets down; (2), testing; (3) in some test mode}	Specifies that the ifAdminStatus variation object cannot be written for the ATM interface, which is identified by the ifType value atm(37). The ifAdminStatus variation object does not support the value testing(3). Max-Access: read-write Access: read-write
ifOperStatus	INTEGER { up(1), down(2), notPresent(6) }	Integer {up (1), ready to pass packets; down(2); testing (3), in some test mode; unknown (4), status can not be determined; dormant (5); notPresent (6), some component is missing; lowerLayerDown (7), down due to state of lower-layer interface(s)}	Specifies that the ifOperStatus variation object cannot detect all states. This variation object does not support the values testing(3), unknown(4), dormant(5), and lowerLayerDown(7). Max-Access: read-only

Table 8-17 Cisco WAN IF Capabilities Used for AXSM (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
ifLastChange	none	`TimeTicks`	Specifies that the value cannot reflect the change in the operational status of the interface. Max-Access: read-only Access: read-only
ifLinkUpDownTrap Enable	none	`Integer {enabled (1), disabled (2)}`	Specifies that the write operation is not supported. The default value is `disabled`. Max-Access: read-write Access: read-only
ifNumber	none	`Integer32`	Specifies that the `ifNumber` variation object is not supported. Max-Access: read-only Access: not implemented
ifAlias	none	`DisplayString (SIZE(0..64))`	Specifies that the `ifAlias` variation object is not supported. Max-Access: read-write Access: not implemented

Cisco IF Capabilities Used for AXSM-E

The `ifGeneralInformationGroup` is the object group applicable to Cisco IF capabilities used for AXSM-E. These capabilities reside in the CISCO-AXSME-IF-CAPABILITY.my file.

For a more detailed description, see Chapter 2, "Standard MIB Objects."

The variation type of MIB objects are listed in Table 8-18.

Table 8-18 Cisco AXSM-E IF Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
ifTableLastChange	none	`TimeTicks`	Specifies that the `ifTableLastChange` variation object is not supported. Access: not-implemented
ifPhysAddress	none	`PhysAddress`	Specifies that the `ifPhysAddress` variation object is not supported. Access: not-implemented
ifAdminStatus	`INTEGER { up(1), down(2) }`	`INTEGER { up(1), -- ready to pass packets down(2), testing(3) -- in some test mode }`	Specifies that the value `testing(3)` is not supported.

Table 8-18 Cisco AXSM-E IF Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
ifOperStatus	INTEGER { up(1), down(2), notPresent(6), lowerLayerDown(7) }	Integer {up (1), ready to pass packets; down(2); testing (3), in some test mode; unknown (4), status can not be determined; dormant (5); notPresent (6), some component is missing; lowerLayerDown (7), down due to state of lower-layer interface(s)}	Specifies that the ifOperStatus variation object is unable to detect all states. The values testing(3), unknown(4), and dormant(5) are not supported.
ifLastChange	none	TimeTicks	Specifies that the ifLastChange variation object always returns a value zero.
ifLinkUpDownTrap Enable	none	INTEGER { enabled(1), disabled(2) }	Specifies that the write operation is not supported. The default value is disabled. Access: read-only
ifNumber	none	Integer32	Specifies that the ifNumber variation object is not supported. Access: not-implemented
ifAlias	none	DisplayString (SIZE(0..64))	Specifies that the ifAlias variation object is not supported. Access: not-implemented

SNMPv2 Capabilities

The following are the object groups applicable to Cisco WAN MGX8800 SNMPv2 capabilities:

- snmpGroup
- systemGroup
- snmpCommunityGroup
- snmpBasicNotificationsGroup

The SNMPv2 capabilities reside in the CISCO-WAN-MGX8800-SNMPV2-CAPABILITY.my file.

For a more detailed description, see Chapter 4, “PXM MIB Objects.”

The variation type of MIB objects are listed in Table 8-19.

Table 8-19 Cisco WAN MGX8800 SNMPv2 Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
sysName	DisplayString (SIZE(1..32))	DisplayString (SIZE (0..255))	Determines that the maximum length of the string is 32. Max-Access: read-write
sysORLastChange	none	TimeStamp	Specifies that the sysORLastChange variation object is not supported. Max-Access: read-only Access: not implemented
sysORID	none	OBJECT IDENTIFIER	Specifies that the sysORID variation object is not supported. Max-Access: read-only Access: not implemented
sysORUpTime	none	TimeStamp	Specifies that the sysORUpTime variation object is not supported. Max-Access: read-only Access: not implemented
sysORDescr	none	DisplayString	Specifies that the sysORDescr variation object is not supported. Max-Access: read-only Access: not implemented

PNNI Capabilities

The following are the object groups applicable to the CISCO-WAN-PNNI-CAPABILITY.my file:

- pnniGeneralMinGroup
- pnniGeneralBorderGroup
- pnniNodeMinGroup
- pnniNodePglMinGroup
- pnniNodePglLgnGroup
- pnniSummaryAddressLgnGroup
- pnniLinkMinGroup
- pnniNodeTimerMinGroup
- pnniNodeSvccLgnGroup
- pnniScopeMinGroup

For a more detailed description, see Chapter 5, “PNNI MIB Objects.”

The variation type of MIB objects are listed in Table 8-20.

Table 8-20 Cisco WAN PNNI Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
pnniNodeDomainName	none	`DisplayString`	Specifies that the `pnniNodeDomainName` variation object is not supported. Max-Access: read-create Access: not implemented
pnniNodeRowStatus	`INTEGER { active(1), createAndGo(4), destroy(6) }`	`RowStatus`	Specifies that the values `notInService(2)`, `createAndWait(5)`, and `notReady(3)` are not supported. The value `active(1)` is not supported for a write operation. Max-Access: read-create
pnniSummaryAddressRowStatus	`INTEGER { active(1), createAndGo(4), destroy(6) }`	`RowStatus`	Specifies that the values `notInService(2)`, `createAndWait(5)`, and `notReady(3)` are not supported. The value `active(1)` is not supported for a write operation. Max-Access: read-create

RS-232 Capabilities

The following are the object groups applicable to the CISCO-WAN-RS-232-CAPABILITY.my file:

- `rs232Group`
- `rs232AsyncGroup`

For a more detailed description, see Chapter 2, "Standard MIB Objects."

The variation type of MIB objects are listed in Table 8-21.

Table 8-21 Cisco WAN RS-232 Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
rs232PortType	`INTEGER { rs232(2) }`	`INTEGER { other(1), rs232(2), rs422(3), rs423(4), v35(5), x21(6) }`	Specifies that the values `other(1)`, `rs422(3)`, `rs423(4)`, `v35(5)`, and `x21(6)` are not supported. Max-Access: read-only
rs232PortInSigNumber	none	`Integer32`	Specifies that the `rs232PortInSigNumber` variation object is not supported. Max-Access: read-only Access: not implemented
rs232PortOutSigNumber	none	`Integer32`	Specifies that the `rs232PortOutSigNumber` variation object is not supported. Max-Access: read-only Access: not implemented

Table 8-21 Cisco WAN RS-232 Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
rs232PortInSpeed	Integer32	Integer32	Supports values 2400, 9600 and 192000. Max-Access: read-write
rs232PortOutSpeed	Integer32	Integer32	Supports values 2400, 9600 and 192000. Max-Access: read-write
rs232PortInFlowType	INTEGER {ctsRts(2) }	INTEGER { none(1), ctsRts(2), dsrDtr(3)}	Specifies that the values none(1) and dsrDtr(3) are not supported. Max-Access: read-write
rs232PortOutFlowType	INTEGER {ctsRts(2) }	INTEGER { none(1), ctsRts(2), dsrDtr(3)}	Specifies that the values none(1) and dsrDtr(3) are not supported. Max-Access: read-write
rs232AsyncPortStopBits	INTEGER {one(1), two(2), oneAndHalf(3) }	INTEGER { one(1), two(2), oneAndHalf(3), dynamic(4) }	Specifies that the value dynamic(4) is not supported. Max-Access: read-write
rs232InSigPortIndex	none	InterfaceIndex	Specifies that the rs232InSigPortIndex variation object is not supported. Max-Access: read-only Access: not implemented
rs232InSigName	none	INTEGER { rts(1), cts(2), dsr(3), dtr(4), ri(5), dcd(6), sq(7), srs(8), srts(9), scts(10), sdcd(11) }	Specifies that the rs232InSigName variation object is not supported. Max-Access: read-only Access: not implemented
rs232InSigState	none	INTEGER { none(1), on(2), off(3) }	Specifies that the rs232InSigState variation object is not supported. Max-Access: read-only Access: not implemented
rs232InSigChanges	none	Counter32	Specifies that the rs232InSigChanges variation object is not supported. Max-Access: read-only Access: not implemented
rs232OutSigPortIndex	none	InterfaceIndex	Specifies that the rs232OutSigPortIndex variation object is not supported. Max-Access: read-only Access: not implemented
rs232OutSigName	none	INTEGER { rts(1), cts(2), dsr(3), dtr(4), ri(5), dcd(6), sq(7), srs(8), srts(9), scts(10), sdcd(11) }	Specifies that the rs232OutSigName variation object is not supported. Max-Access: read-only Access: not implemented

Table 8-21 Cisco WAN RS-232 Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
rs232OutSigState	none	`INTEGER { none(1), on(2), off(3) }`	Specifies that the `rs232OutSigState` variation object is not supported. Max-Access: read-only Access: not implemented
rs232OutSigChanges	none	`Counter32`	Specifies that the `rs232OutSigChanges` variation object is not supported. Max-Access: read-only Access: not implemented
rs232AsyncPortStopBits	`INTEGER { one(1), two(2), oneAndHalf(3) }`	`INTEGER {one(1), two(2), oneAndHalf(3), dynamic(4) }`	Specifies that the `dynamic(4)` value is not supported. Max-Access: read-write
rs232AsyncPortAutobaud	none	`INTEGER { enabled(1), disabled(2) }`	Specifies that write access is not supported. Max-Access: read-write Access: read-only
rs232AsyncPortParityErrs	none	`Counter32`	Specifies that the `rs232AsyncPortParityErrs` variation object is not supported. Max-Access: read-only Access: not implemented
rs232AsyncPortFraming Errs	none	`Counter32`	Specifies that the `rs232AsyncPortFramingErrs` variation object is not supported. Max-Access: read-only Access: not implemented
rs232AsyncPortOverrun Errs	none	`Counter32`	Specifies that the `rs232AsyncPortOverrunErrs` variation object is not supported. Max-Access: read-only Access: not implemented

Cisco WAN Resource Partition Capabilities

The following are the object groups applicable to the CISCO-WAN-RSRC-PART-CAPABILITY.my file:

- cwRsrcPartMIBGroup
- cwRsrcPartIlmiMIBGroup

For a more detailed description, see Chapter 3, "Cisco Enterprise MIB Objects."

The variation type of MIB objects are listed in Table 8-22.

Table 8-22 Cisco WAN Resource Partition Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
cwRsrcPartID	Unsigned32(1..20)	Unsigned32 (1.. 255)	Supports the maximum value of 20. Max-Access: not-accessible
cwRsrcPartController	Unsigned32(1..20)	Unsigned32 (1.. 255)	Supports the maximum value of 20. Max-Access: not-accessible
cwRsrcPartRowStatus	INTEGER { active(1), notInService(2), createAndGo(4), destroy(6)}	RowStatus	Specifies that the value createAndWait(5) is not supported. Max-Access: read-create

Cisco WAN Resource Partition Capabilities for RPM

This section describes the agent capabilities that support the Cisco WAN Resource Partition MIB for RPM.

The cwRsrcPartMIBGroup is the object group applicable to the CISCO-WAN-RPM-RSRC-PART-CAPABILITY.my file.

For a more detailed description, seeChapter 3, "Cisco Enterprise MIB Objects."

The variation type of MIB objects are listed in Table 8-23.

Table 8-23 Cisco WAN Resource Partition Capabilities for RPM

Variation Name	Capability Syntax	MIB Syntax	Description
cwRsrcPartID	Unsigned32 (1..10)	Unsigned32 (1.. 255)	Specifies that a limited range is supported. The value 1 is reserved for the PNNI controller.
cwRsrcPartController	none	Unsigned32 (1.. 255)	Specifies that the value 2 is assumed to be the PNNI controller.
cwRsrcPartEgrGuarPct BwConf	none	Unsigned32 (0..1000000)	Specifies that the cwRsrcPartEgrGuarPctBwConf variation object supports only a whole percentage, for example, 1 percent, 10 percent, and so forth.

Table 8-23 Cisco WAN Resource Partition Capabilities for RPM (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
cwRsrcPartEgrMaxPctBwConf	none	`Unsigned32 (0..1000000)`	Specifies that the `cwRsrcPartEgrMaxPctBwConf` variation object supports only a whole percentage, for example, 1 percent, 10 percent, and so forth.
cwRsrcPartIngGuarPctBwConf	none	`Unsigned32 (0..1000000)`	Specifies that the `cwRsrcPartIngGuarPctBwConf` variation object supports only a whole percentage, for example, 1 percent, 10 percent, and so forth.
cwRsrcPartIngMaxPctBwConf	none	`Unsigned32 (0..1000000)`	Specifies that the `cwRsrcPartIngMaxPctBwConf` variation object supports only a whole percentage, for example, 1 percent, 10 percent, and so forth.
cwRsrcPartVpiLo	`Unsigned32 (0..255)`	`Unsigned32 (0..4095)`	Supports the restricted range. For VCC partition, the value 0 is supported. For VPC partition, the range is from 1 to 255.
cwRsrcPartVpiHigh	`Unsigned32 (0..255)`	`Unsigned32 (0..4095)`	Supports the restricted range. For a VCC partition, the value 0 is supported. For a VPC partition, the valid range is from 1 to 255.
cwRsrcPartVciLo	`Unsigned32 (1..65535)`	`Unsigned32 (0..65535)`	Supports the restricted range. For a VCC partition, the valid range is from 1 to 3808. For a VPC partition, the valid range is from 0 to 65535.
cwRsrcPartVciHigh	`Unsigned32 (1..65535)`	`Unsigned32 (0..65535)`	Supports the restricted range. For a VCC partition, the valid range is from 1 to 3808. For a VPC partition, the valid range is from 0 to 65535.
cwRsrcPartGuarCon	none	`Unsigned32 (0..131072)`	Specifies that write operation is not supported. Access: read-only
cwRsrcPartMaxCon	none	`Unsigned32 (0..131072)`	Specifies that write operation is not supported. Access: read-only
cwRsrcPartRowStatus	`INTEGER { active(1), createAndGo(4), destroy(6) }`	`RowStatus`	Specifies that the `cwRsrcPartRowStatus` variation object does not support `notInService(2)`, `notReady(3)`, and `createAndWait(5)`.

SONET Capabilities

The following are the object groups applicable to the CISCO-WAN-SONET-CAPABILITY.my file:

- sonetMediumStuff2
- sonetSectionStuff2
- sonetLineStuff2
- sonetPathStuff2
- sonetFarEndLineStuff2
- sonetFarEndPathStuff2

For a more detailed description, see Chapter 2, "Standard MIB Objects."

The variation type of MIB objects are listed in Table 8-24.

Table 8-24 Cisco WAN SONET Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
sonetPathCurrentWidth	none	INTEGER { sts1(1), sts3cSTM1(2), sts12cSTM4(3), sts24c(4), sts48cSTM16(5) }	Specifies that write access is not supported. Max-Access: read-write Access: read-only
sonetMediumLineCoding	none	INTEGER {sonetMediumOther(1), sonetMediumB3ZS(2), sonetMediumCMI(3), sonetMediumNRZ(4), sonetMediumRZ(5) }	Specifies that write access is not supported. Max-Access: read-write Access: read-only
sonetMediumLineType	none	INTEGER {sonetOther(1), sonetShortSingleMode(2), sonetLongSingleMode(3), sonetMultiMode(4), sonetCoax(5), sonetUTP(6)}	Specifies that write access is not supported. Max-Access: read-write Access: read-only
sonetMediumCircuit Identifier	none	DisplayString (SIZE (0..255))	Specifies that write access is not supported. Max-Access: read-write Access: read-only
sonetMediumLoopback Config	none	BITS {sonetNoLoop(0), sonetFacilityLoop(1), sonetTerminalLoop(2), sonetOtherLoop(3)}	Specifies that the variation object is not supported. Max-Access: read-write Access: not implemented

Cisco WAN SVC Capabilities

The following are the object groups applicable to the CISCO-WAN-SVC-CAPABILITY.my file:

- cwsInfoGrp
- cwspConfigGrp
- cwspCacConfigGrp
- cwspCallStatsGrp
- cwspSigStatsGrp
- cwspCallGrp
- cwspAbrCallGrp
- cwspLoadGrp
- cwspAddressGrp
- cwspConnTraceGrp
- cwspOperationGrp
- cwspSpvcGrp

For a more detailed description, see Chapter 5, "PNNI MIB Objects."

The variation type of MIB objects are listed in Table 8-25.

Table 8-25 Cisco WAN SVC Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
cwspConfigTableRowStatus	INTEGER {active(1), createAndGo(4), destroy(6)}	RowStatus	Specifies that the values notInService(2), notReady(3), and createAndWait(5) are not supported. Max-Access: read-create
cwspAddressRowStatus	INTEGER {active(1), createAndGo(4), destroy(6)}	RowStatus	Specifies that the values notInService(2) and createAndWait(5) are not supported. The value active(1) cannot beset. Max-Access: read-create
cwspConnTraceQueryStatus	INTEGER {active(1), createAndGo(4), destroy(6)}	RowStatus	Specifies that the values notInService(), notReady(3), and createAndWait(5)are not supported. The value active(1) cannot be set. Max-Access: read-create
cwspSpvcNodePrefix	none	none	Specifies that the write operation is not supported. Access: read-only
cwspMaxP2pCalls	none	Integer32 (0..65535)	Specifies that the write operation is not supported. Max-Access: read-create Access: read-only

Cisco System Capabilities

This section describes the agent capabilities that support the Cisco System MIB.

The following are the object groups applicable to the CISCO-WAN-SYSTEM-CAPABILITY.my file:

- ciscoSystemClockGroup
- ciscoSystemLocationGroup

The agent capabilities for Cisco System does not contain any variation objects.

Cisco AXSM Cisco SONET Capabilities

This section describes the agent capabilities that support the Cisco SONET MIB.

The following are the object groups applicable to the CISCO-AXSM-CISCO-SONET-CAPABILITY.my file:

- ciscoSonetConfMIBGroup
- ciscoSonetStatsMIBGroup
- ciscoSonetTraceMIBGroup
- ciscoSonetApsMIBGroup

For a more detailed description, see Chapter 3, "Cisco Enterprise MIB Objects."

The variation type of MIB objects are listed in Table 8-26.

Table 8-26 Cisco AXSM Cisco SONET Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
csConfigFrame Scramble	INTEGER { enabled(2) }	INTEGER {disabled(1), enabled(2)}	Specifies the value disabled(1) is not supported.
csApsWaitToRestore	Unsigned32(5..12)	Unsigned32 (1..12) in units of minutes	Determines the wait to restore time interval is between 5 to 12 minutes.
csApsArchMode	INTEGER { onePlusOne(1), oneToOne(2)}	INTEGER {onePlusOne(1), oneToOne(2), anexBOnePlusOne(3), ycableOnePlusOneNok1k2(4), straightOnePlusOneNok1k2(5)}	Specifies the values anexBOnePlusOne(3), ycableOnePlusOneNok1k2(4), straightOnePlusOneNok1k2(5) are not supported.
cssTraceReceived	none	OCTET STRING (SIZE (0|16|64))	Specifies the cssTraceReceived variation object is not supported. Access: not implemented
cspTraceReceived	none	OCTET STRING (SIZE(0|16|64))	Specifies the cspTraceReceived variation object is not supported. Access: not implemented

Cisco AXSM-E Cisco SONET Capabilities

The following are the object groups applicable to the CISCO-AXSME-CISCO-SONET-CAPABILITY.my file for AXSM-E:

- ciscoSonetConfMIBGroup
- ciscoSonetStatsMIBGroup
- ciscoSonetTraceMIBGroup
- ciscoSonetApsMIBGroup

For a more detailed description, see Chapter 3, "Cisco Enterprise MIB Objects."

The variation type of MIB objects are listed in Table 8-27.

Table 8-27 Cisco AXSM-E Cisco SONET Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
csConfigFrame Scramble	INTEGER { enabled(2) }	INTEGER { disabled(1), enabled(2) }	Specifies that the value disabled(1) is not supported.
csApsArchMode	INTEGER { onePlusOne(1), oneToOne(2), anexBOnePlusOne(3), straightOnePlusOne Nok1k2(5) }	INTEGER { onePlusOne(1), oneToOne(2), anexBOnePlusOne(3), ycableOnePlusOneNok1k2(4), straightOnePlusOneNok1k2(5) }	Specifies that the value ycableOnePlusOneNok1k2(4) is not supported.
csApsDirection	none	INTEGER { uniDirectional(1), biDirectional(2) }	Specifies that the value uniDirectional(1) is ignored for the following scenarios: • csApsArchMode equals to annexBOnePlusOne(3) • csApsChannelProtocol equals to itu(2)
csApsRevertive	none	INTEGER { nonrevertive(1), revertive(2) }	Specifies that the value revertive(2) is ignored for the following scenarios: • csApsArchMode equals to annexBOnePlusOne(3) • csApsChannelProtocol equals to itu(2)

Table 8-27 Cisco AXSM-E Cisco SONET Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
cssTraceReceived	none	`OCTET STRING (SIZE(0|16|64))`	Specifies that the `cssTraceReceived` variation object is not supported. Access: not-implemented
cspTraceReceived	none	`OCTET STRING (SIZE(0|16|64))`	Specifies that the `cspTraceReceived` variation object is not supported. Access: not-implemented

Cisco AXSM WAN Module Capabilities

This section describes the agent capabilities that support the Cisco WAN Module MIB.

The following are the object groups applicable to the CISCO-AXSM-WAN-MODULE-CAPABILITY.my file:

- `cwmConfigGroup`
- `ConfigGroup`

For a more detailed description, see Chapter 3, "Cisco Enterprise MIB Objects."

The variation type of MIB objects are listed in Table 8-28.

Table 8-28 Cisco AXSM WAN Module Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
cwmIngressSCTFileName	none	`DisplayString`	Specifies the `cwmIngressSCTFileName` variation object is not supported. Access: not implemented
cwmAutoLineDiagEnable	none	`INTEGER { enable(1), disable(2) }`	Specifies the `cwmAutoLineDiagEnable` variation object is not supported. Access: not implemented
cwmStatBucketInterval	none	`INTEGER { five(5), ten(10), fifteen(15), twenty(20), thirty(30), sixty(60) }`	Sets the default to value `fifteen(15)`.
cwmStatCollectionInterval	none	`INTEGER { default(0), one(1), five(5) }`	Specifies the `cwmStatCollectionInterval` variation object is not supported. Access: not implemented
cwmStatCollectionStatus	none	`INTEGER { enable(1), disable(2) }`	Specifies the `cwmStatCollectionStatus` variation object is not supported. Access: not implemented
cwmStatCurrentLevel	none	`StatisticsLevel`	Specifies the `cwmStatCurrentLevel` variation object is not supported. Access: not implemented

Table 8-28 Cisco AXSM WAN Module Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
cwmStatLevelConfigured	none	`StatisticsLevel`	Specifies the `cwmStatLevelConfigured` variation object is not supported. Access: not implemented
cwmStatMaximum Connections	none	`Unsigned32`	Specifies the `cwmStatMaximumConnections` variation object is not supported. Access: not implemented

Cisco AXSM-E WAN Module Capabilities

This section describes the agent capabilities that support the Cisco WAN Module MIB for AXSM-E.

The following are the object groups applicable to the CISCO-AXSME-WAN-MODULE-CAPABILITY.my file:

- `cwmConfigGroup`
- `cwmStatConfigGroup`

For a more detailed description, see Chapter 3, "Cisco Enterprise MIB Objects."

The variation type of MIB objects are listed in Table 8-29.

Table 8-29 Cisco AXSM-E WAN Module Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
cwmIngressSCTFileName	none	`DisplayString`	Specifies that the `cwmIngressSCTFileName` variation object is not supported. Access: not-implemented
cwmStatCollectionStatus	none	`INTEGER { enable(1), disable(2) }`	Specifies that the `cwmStatCollectionStatus` variation object is not supported. Access: not-implemented

Cisco AXSM-E ATM Cell Layer Capabilities

This section describes the agent capabilities that support the Cisco ATM Cell Layer MIB for AXSM-E.

The following are the object groups applicable to the CISCO-AXSME-ATM-CELL-LAYER-CAPABILITY.my file:

- `caclATMSwitchStatsMIBGroup`
- `caclATMEndSyatemStatsMIBGroup`
- `caclIntervalStatsMIBGroup`

For a more detailed description, see Chapter 3, "Cisco Enterprise MIB Objects."

The variation type of objects are listed in Table 8-30.

Table 8-30 Cisco AXSM-E ATM Cell Layer Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
caclInXmtCLP0Cells	none	Counter32	Specifies the caclInXmtCLP0Cells variation object is not supported. Access: not-implemented
caclInXmtCLP1Cells	none	Counter32	Specifies the caclInXmtCLP1Cells variation object is not supported. Access: not-implemented
caclInUpcCLP0DiscCells	none	Counter32	Specifies the caclInUpcCLP0DiscCells variation object is not supported. Access: not-implemented
caclInUpcTotalDiscCells	none	Counter32	Specifies the caclInUpcTotalDiscCells variation object is not supported. Access: not-implemented
caclInUpcTotalNonComp Cells	none	Counter32	Specifies the caclInUpcTotalNonCompCells variation object is not supported. Access: not-implemented
caclOutXmtCLP0Cells	none	Counter32	Specifies the caclOutXmtCLP0Cells variation object is not supported. Access: not-implemented
caclOutXmtCLP1Cells	none	Counter32	Specifies the caclOutXmtCLP1Cells variation object is not supported. Access: not-implemented
caclOutRcvIdleCells	none	Counter32	Specifies the caclOutRcvIdleCells variation object is not supported. Access: not-implemented
caclOutVpiVciErrCells	none	Counter32	Specifies the caclOutVpiVciErrCells variation object is not supported. Access: not-implemented
caclIntervalInXmtCLP0 Cells	none	Gauge32	Specifies the caclIntervalInXmtCLP0Cells variation object is not supported. Access: not-implemented

Table 8-30 Cisco AXSM-E ATM Cell Layer Capabilities (continued)

Variation Name	Capability Syntax	MIB Syntax	Description
caclIntervalInXmtCLP1 Cells	none	`Gauge32`	Specifies the `caclIntervalInXmtCLP1Cells` variation object is not supported. Access: not-implemented
caclIntervalInUpcCLP0 DiscCells	none	`Gauge32`	Specifies the `caclIntervalInUpcCLP0Disc Cells` variation object is not supported. Access: not-implemented
caclIntervalInUpcTotalDisc Cells	none	`Gauge32`	Specifies the `caclIntervalInUpcTotalDisc Cells` variation object is not supported. Access: not-implemented
caclIntervalInUpcTotNon CmpCells	none	`Gauge32`	Specifies the `caclIntervalInUpcTotNonCmp Cells` variation object is not supported. Access: not-implemented
caclIntervalOutXmtCLP0 Cells	none	`Gauge32`	Specifies the `caclIntervalOutXmtCLP0Cells` variation object is not supported. Access: not-implemented
caclIntervalOutXmtCLP1 Cells	none	`Gauge32`	Specifies the `caclIntervalOutXmtCLP1Cells` variation object is not supported. Access: not-implemented
caclIntervalOutRcvIdle Cells	none	`Gauge32`	Specifies the `caclIntervalOutRcvIdleCells` variation is not supported. Access: not-implemented
caclIntervalOutVpiVciErr Cells	none	`Gauge32`	Specifies the `caclIntervalOutVpiVciErrCells` variation is not supported. Access: not-implemented

Cisco AXSM-E ATM Connection Statistics Capabilities

This section describes the agent capabilities that support the Cisco WAN ATM Connection Statistics MIB for AXSM-E.

The `cwacsGroup` is the object groups applicable to the CISCO-AXSME-ATM-CONN-STAT-CAPABILITY.my file.

For a more detailed description, see Chapter 3, "Cisco Enterprise MIB Objects."

The variation type of objects are listed in Table 8-31.

Table 8-31 Cisco AXSM-E ATM Connection Statistics Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
cwacsIngRcvEFCI0	none	`Counter32`	Specifies that the `cwacsIngRcvEFCI0` variation object is not supported. Access: not-implemented
cwacsIngEFCI0Discard	none	`Counter32`	Specifies that the `cwacsIngEFCI0Discard` variation object is not supported. Access: not-implemented
cwacsIngXmtFRm	none	`Counter32`	Specifies that the `cwacsIngXmtFRm` variation object is not supported. Access: not-implemented
cwacsIngXmtBRmFsRm	none	`Counter32`	Specifies that the `cwacsIngXmtBRmFsRm` variation object is not supported. Access: not-implemented
cwacsEgrRcvEFCI0	none	`Counter32`	Specifies that the `cwacsEgrRcvEFCI0` variation object is not supported. Access: not-implemented
cwacsEgrEFCI0Discard	none	`Counter32`	Specifies that the `cwacsEgrEFCI0Discard` variation object is not supported. Access: not-implemented
cwacsEgrXmtFRm	none	`Counter32`	Specifies that the `cwacsEgrXmtFRm` variation object is not supported. Access: not-implemented
cwacsEgrXmtBRmFsRm	none	`Counter32`	Specifies that the `cwacsEgrXmtBRmFsRm` variation object is not supported. Access: not-implemented
cwacsEgrRcvEOF1	none	`Counter32`	Specifies that the `cwacsEgrRcvEOF1` variation object is not supported. Access: not-implemented

Cisco WAN RPM Subinterface Capabilities

This section describes the agent capabilities that support the Cisco WAN RPM Subinterface MIB for RPM.

The `ciscoWanRpmSubIfMIBGroup2` is the object group applicable to the CISCO-WAN-RPM-SUBIF-CAPABILITY.my file.

The variation type of MIB objects are listed in Table 8-32.

Table 8-32 Cisco WAN RPM Subinterface Capabilities

Variation Name	Capability Syntax	MIB Syntax	Description
cwrSubIfAdapterIf	`Unsigned32(1..1)`	`Unsigned32`	Supports value 1 only.
cwrSubIfRowStatus	`INTEGER { active(1), notInService(2), createAndGo(4), destroy(6) }`	`RowStatus`	Specifies that the `cwrSubIfRowStatus` variation object does not support `notReady(3)` and `createAndWait(5)`.

CHAPTER 9

Statistics Supported

This chapter describes the statistics supported by the AXSM MIBs.

Contents of this chapter include:

- Line Statistics
  - Ingress Per Line Physical Layer Statistics before Policer
  - Ingress Per Line Policing Statistics
  - Egress Per Line (PHY) Statistics to Port
- Port Statistics
  - Ingress Port Statistics
  - Egress IF Statistics

Line Statistics

Line Statistics are supported primarily by the physical devices. In addition, the ATLAS policer is available only on AXSM cards with OC12, OC3, T3, and E3 physical interfaces.

Note The ATLAS policier is not available on AXSM cards with a OC48 physical interface.

The ATLAS is used in both the ingress and egress directions.

Ingress Per Line Physical Layer Statistics before Policer

Table 9-1 shows the ingress per line statistics Physical Layer (PHY) before the policer for the various AXSM options.

Table 9-1 Ingress per Line (PHY) Statistics before Policer

Statistic	BXM	AXSM OC48	AXSM OC3/12	AXSM T3/E3	MIB Object
CLP0 Cells From Port	yes	no	yes	yes	`caclInRcvCLP0Cells`
CLP1 Cells From Port	yes	no	yes	yes	`caclInRcvCLP1Cells`

Table 9-1 Ingress per Line (PHY) Statistics before Policer (continued)

Statistic	BXM	AXSM OC48	AXSM OC3/12	AXSM T3/E3	MIB Object
Total Valid OAM Cells	no	no	yes	yes	`caclInValidOAMCells`
Total Valid RM Cells	no	no	yes	yes	none
Total Errored OAM/RM Cells	no	no	yes	yes	`caclInErrOAMCells`
non-zero GFC Cells	no	no	yes	yes	`caclInGfcCells`
Last Unknown VPI.VCI (invalid)	yes	no	yes	yes	`caclInLastUnknVpi`, `caclInLastUnknVci`
HEC Errored Cells Discarded	yes	yes	yes	yes	`caclInHecErrDiscCells`
HEC Errored Cells Corrected	yes	yes	yes	yes	`caclInHecErrCorrectedCells`

Ingress Per Line Policing Statistics

The ingress per line policing statistics (PHY) are shown in Table 9-2 for the various AXSM options.

Table 9-2 Ingress per Line Policing Statistics—AXSM Options

Statistic	BXM	AXSM OC48	AXSM OC3/12	AXSM T3/E3	MIB Object
UPC CLP0 Discards	no	no	yes	yes	`caclInUpcCLP0DiscCells`
UPC CLP0+1 Discards	yes	no	yes	yes	`caclInUpcTotalDiscCells`
Non—Compliant CLP0+1	no	no	yes	yes	`caclInUpcTotalNonCompCells`

Egress Per Line (PHY) Statistics to Port

The egress per line statistics to port are shown in Table 9-3 for the various AXSM options.

Table 9-3 Egress Per Line (PHY) Statistics to Port—AXSM Options

Statistic	BXM	AXSM OC48	AXSM OC3/12	AXSM T3/E3	MIB Object
CLP0 Cells to Port	yes	no	yes	yes	`caclOutXmtCLP0Cells`
CLP1 Cells to Port	yes	no	yes	yes	`caclOutXmtCLP1Cells`
CLP0+1 Cells to Port	yes	yes[1]	yes	yes	none
OAM Valid Cells	no	no	yes	yes	`caclOutValidOAMCells`
RM Valid Cells	no	no	yes	yes	`caclOutRcvValidRMCells`
OAM/RM Errored Cells	no	no	yes	yes	`caclOutErrOAMCells`

1. This statistic is specified from phy non idle

Port Statistics

Port Statistics refer to the statistics collected on a logical port (not physical port) in both ingress and egress directions.

Ingress Port Statistics

Table 9-4 shows the ingress port statistics for the various AXSM options.

Note For OAM cell statistics, the QE48 chip supports all 64 Service Groups.

Table 9-4 Ingress Port Statistics for AXSM Options

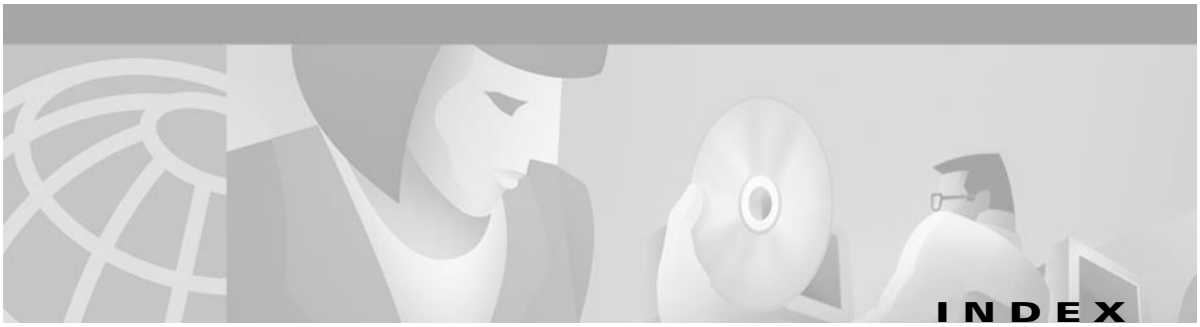
Statistic	BXM	AXSM OC48	AXSM OC3/12	AXSM T3/E3	MIB Object
CLP0 cells from Policer	yes	yes	yes	yes	caviEgrRcvClp0Cells
CLP1 cells from Policer	yes	yes	yes	yes	caviEgrRcvClp1Cells
CLP0 cells discarded (dropped)	yes	yes	yes	yes	caviEgrClp0DiscCells
CLP1 cells discarded (dropped)	yes	yes	yes	yes	caviEgrClp1DiscCells
CLP0 cells to backplane	no	yes	yes	yes	caviEgrXmtClp0Cells
CLP1 cells to backplane	no	yes	yes	yes	caviEgrXmtClp1Cells
Total OAM cells	yes	no	no	no	caviEgrRcvOAMCells

Egress IF Statistics

Table 9-5 shows the egress IF statistics for the various AXSM options.

Table 9-5 Egress IF Statistics for AXSM Options

Statistic	BXM	AXSM -OC48	AXSM OC3/12	AXSM T3/E3	MIB Object
CLP0 cells from backplane	yes	yes	yes	yes	caviIngRcvClp0Cells
CLP1 cells from backplane	yes	yes	yes	yes	caviIngRcvClp1Cells
CLP0 cells discarded (dropped)	yes	yes	yes	yes	caviIngClp0DiscCells
CLP1 cells discarded (dropped)	yes	yes	yes	yes	caviIngClp1DiscCells
CLP0 cells to port	no	yes	yes	yes	caviIngXmtClp0Cells
CLP1 cells to port	no	yes	yes	yes	caviIngXmtClp1Cells
Total OAM cells	yes	yes	yes	yes	caviIngXmtOAMCells



A

B

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y