Cisco RF Gateway 10 MIB Specifications Guide

Cisco IOS Release 12.2(50)SQ and Earlier Releases
Cisco IOS-XE Release 3.4.1SQ and Earlier Releases

October 2014

Cisco Systems, Inc.
www.cisco.com

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco RF Gateway 10 MIB Specifications Guide

CONTENTS

Preface

This guide describes the implementation of the Simple Network Management Protocol (SNMP) and Management Information Base (MIB) for Cisco RF Gateway-10 (RFGW-10).

SNMP provides a set of commands for setting and retrieving the values of operating parameters on the RFGW-10. RFGW-10 information is stored in a virtual storage area called a MIB, which contains many objects that describe RFGW-10 components and provides information about the status of the components. This chapter provides an overview of this guide with the following sections:

Document Revision History

This guide contains MIB information of:

- Cisco IOS Release 12.2(50)SQ
- Cisco IOS-XE Release 3.4.1SQ
- All earlier releases

For MIB information of later releases, use the Cisco MIB Locater tool.

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

Release	Part Number	Publication Date	Change Summary
Cisco IOS-XE Release 3.4.1SQ	—	October 2014	Updated SCTE-HMS-QAM-MIB
Cisco IOS-XE Release 3.4.0SQ	OL-18688-06	June 2014	Updated SCTE-HMS-MPEG-MIB
Cisco IOS-XE Release 3.3.1SQ	OL-18688-05	November 2013	Added CISCO-ENTITY-REDUNDANCY-MIB
Cisco IOS-XE Release 3.3.0SQ	OL-18688-04	July 2013	Updated SCTE-HMS-MPEG-MIB
Cisco IOS-XE Release 3.2.0SQ	OL-18688-03	October 2012	Added DOCS-DRF-MIB
Cisco IOS Release 12.2(50)SQ	OL-18688-02	November 2009	Updated SCTE-HMS-MPEG-MIB

Audience

This guide is intended for system and network administrators who must configure the RFGW-10 for operation and monitor its performance in the network.

This guide may also be useful for application developers who are developing management applications for the RFGW-10.

Organization

This guide contains the following chapters:

- Chapter 1, "Cisco RFGW-10 MIB Overview," provides background information about SNMP and its implementation on Cisco RFGW-10.
- Chapter 2, "Configuring SNMP and MIB Support," provides instructions for configuring SNMP management support on the RFGW-10.
- Chapter 3, "MIB Specifications," describes each MIB included in the software image. Each description lists any constraints as to how the MIB is implemented on the RFGW-10.
- Chapter 4, "Monitoring Notifications," describes the SNMP traps and notifications supported by the RFGW-10.
- Chapter A, "Using Cisco RFGW-10 MIBs," describes how to perform common tasks on the RFGW-10.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see *What's New in Cisco Product Documentation* at: http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html.

Subscribe to *What's New in Cisco Product Documentation*, which lists all new and revised Cisco technical documentation, as an RSS feed and deliver content directly to your desktop using a reader application. The RSS feeds are a free service.

Cisco RFGW-10 MIB Overview

This chapter provides an overview of the Cisco RFGW-10. This chapter contains the following topics:

MIB Description

A Management Information Base (MIB) is a collection of information that can be managed by the SNMP manager. The objects in a MIB are organized and identified by object identifiers (OID) that are defined by the IETF and other organizations. The Cisco implementation of SNMP uses MIBs that conform to the MIB II definition that is described in RFC 1213.

Objects can refer to a physical device (such as a line card or clock card or shared port adapter), a software parameter (such as an IP address or operation mode), or a run-time statistic (such as number of packets passed or temperature). When the device contains multiple objects of the same type, it appends a unique instance number to the end of the OID, so that the SNMP manager and agent can distinguish between the different objects.

MIBs can contain two types of managed objects:

- Scalar objects—Define a single object instance (for example, ifNumber in the IF-MIB and bgpVersion in the BGP4-MIB).
- Tabular objects—Define multiple related object instances that are grouped together in MIB tables (for example, ifTable in the IF-MIB defines the interfaces on the RFGW-10). Each row in a MIB table describes all of the parameters for a particular object (such as IP address, clock speed, number of ports, and so forth). SNMP managers can read or set all of the information in a row with one request.

  Typically, each row in a table is identified by a unique index number. Depending on the table, this index either could reflect a physical attribute (such as the slot number in a chassis or port number on a card) or it could be an arbitrary number (such as is used for tables that list error messages or packet statistics).

Each row also has a status object that shows whether the row is created, activated, deactivated, or deleted. When an SNMP manager creates a new row, it typically sets the row's status to create and then populates the row with the desired parameters. The SNMP agent does not use the objects in a row until the SNMP manager sets the row's status to activate. This ensures that the SNMP agent does not try to use a row's parameters until the SNMP manager has finished creating the row and entered all of the row's required parameters.

Benefits of MIB Enhancements

The Cisco RF Gateway-10 router enhanced management feature allows theCisco RFGW-10 router to be managed through the Simple Network Management Protocol (SNMP). The feature also expands the number of Management Information Bases (MIBs) included with the Cisco RFGW-10 router.

Using the Cisco RF Gateway-10 router enhanced management feature, you can:

- Manage and monitor Cisco RFGW-10 resources through an SNMP-based network management system (NMS)
- Use SNMP **set** and **get** requests to access information in Cisco RF Gateway-10 router MIBs
- Reduce the amount of time and system resources required to perform functions such as inventory management

Other benefits include:

- A standards-based technology (SNMP) for monitoring faults and performance on the RFGW-10.
- Support for all SNMP versions (SNMPv1, SNMPv2c, and SNMPv3)
- Notification of faults, alarms, and conditions that might affect services
- A way to access RFGW-10 information other than through the command line interface (CLI)

MIB Dependencies

The SNMP specifications define MIBs in a highly structured hierarchical format, in which MIBs that are lower in the hierarchy use objects that are defined by MIBs higher up in the hierarchy. Each MIB includes a section titled "IMPORTS" that lists the objects it uses that are defined by other MIBs.

For example, the IF-MIB, which defines standard objects for RFGW-10 interfaces, uses the following IMPORT block:

```
IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, Counter32, Gauge32, Counter64,
    Integer32, TimeTicks, mib-2,
    NOTIFICATION-TYPE                       FROM SNMPv2-SMI
    TEXTUAL-CONVENTION, DisplayString,
    PhysAddress, TruthValue, RowStatus,
    TimeStamp, AutonomousType, TestAndIncr   FROM SNMPv2-TC
    MODULE-COMPLIANCE, OBJECT-GROUP,
    NOTIFICATION-GROUP                      FROM SNMPv2-CONF
    snmpTraps                               FROM SNMPv2-MIB
    IANAifType                              FROM IANAifType-MIB;
```

This section shows that the IF-MIB uses objects that are defined by the SNMPv2-SMI, SNMPv2-TC, SNMPv2-CONF, SNMPv2-MIB, and IANAifType-MIB MIBs. To use the IF-MIB with your SNMP management software, you must load these other MIBs as well.

Typically, most SNMP managers use the IMPORT blocks in the MIBs to automatically determine the order in which the MIBs must be loaded. However, if you are manually loading MIBs, you must do so in the proper order.

To determine the dependencies among MIBs, you can use the "View and Download MIBs" tool, which is part of the SNMP Object Navigator on the Cisco MIB Tools page. This URL takes you to the MIB Locator:

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

Link to MIB Locator tool: http://tools.cisco.com/ITDIT/MIBS/MainServlet

MIB Types

MIBs on the Cisco RFGW-10 can be arranged in the following categories:

- SNMP standard MIBs—Part of the SNMPv1, SNMPv2c, and SNMPv3 specifications and must be supported by any agent supporting SNMP network management. These MIBs provide the framework for SNMP management, defining common objects and interfaces.
- Internet standard MIBs—Provide generic definitions for objects that provide information about commonly used protocols, such as IP, TCP, and Internet Control Message Protocol (ICMP). These MIBs are typically defined by the IETF as Internet-Drafts and Request for Comments (RFCs).
- Cisco platform and network-layer enterprise MIBs—Provide information that is specific to Cisco platforms. These MIBs can extend standard MIBs by providing additional related information, or they can provide information about features that are specific to Cisco platforms. Typically, the same Cisco-specific MIB is used on all Cisco platforms that implement the MIB's particular feature. These MIBs are also typically updated whenever the related feature is updated in the Cisco IOS or IOS-XE software.
- Industry standard specific MIBs— Provide information specific to Cable and Video industry, defined by CableLab and SCTE standard body.
- Deprecated MIBs—Supported in earlier releases of Cisco IOS or IOS-XE software but have been replaced by more standardized, scalable MIBs. Network management applications and scripts should convert to the replacement MIBs as soon as possible, because deprecated MIBs could be removed without notice.

Object Identifiers

An object identifier (OID) uniquely identifies a MIB object on a managed RFGW-10 or other network device. All OIDs are arranged in a hierarchical order, with top-level OIDs assigned by standards organizations such as IETF, ISO, and ITU. Lower-level OIDs are assigned by individual vendor organizations, such as Cisco Systems.

Each level in an OID is assigned both a number and a name. The hierarchical structure of the OIDs allow for easy translation between the number and name forms of an OID.

For example, SNMP standard MIBs that are intended for use by all vendors typically start with "1.3.6.1.2.1", which translates as follows:

iso(1).org(3).dod(6).internet(1).mgmt(2).mib-2(1)

Typically, vendor-specific MIBs have OIDs that start with "1.3.6.1.4.1", which translates as follows:

iso(1).org(3).dod(6).internet(1).private(4).enterprises(1)

Cisco Systems was assigned the next OID of "9", so most OIDs for items that are specific to Cisco platforms start with "1.3.6.1.4.1.9":

iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).cisco(9)

For illustrative purposes, the OIDs above are shown with both number and name forms combined. Typically, only the name or number for a level is used. However, names and numbers can be mixed in the same OID. For example, the top-most Cisco-specific OID could also be given as either "1.3.6.1.4.1.cisco" or "iso.org.dod.internet.private.enterprises.9".

To translate OIDs between their name and number format, and to display the location of any OID in the OID tree, you can use the SNMP Object Navigator on the Cisco MIB Tools page. This URL takes you to the MIB Locator:

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

For a listing of all of the objects and OIDs that are included in any particular MIB, you can download the text files at the following URL:

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

Link to MIB Locator tool: http://tools.cisco.com/ITDIT/MIBS/MainServlet

SNMP Overview

The Cisco RFGW-10 can be managed through SNMP, which is an application-layer protocol that provides a standardized framework and a common language for monitoring and managing devices in a network. The SNMP framework has the following main parts:

- An SNMP manager—A system used to control and monitor the activities of network hosts by using SNMP commands. The most common managing system is called a network management system (NMS), which can be either a standalone device that is dedicated to network management, or a workstation that is running network management applications. Many network management applications are available and range from simple, freely available command-line applications to feature-rich, commercial products with sophisticated graphical user interfaces.
- An SNMP agent—A software component in a managed device that maintains the SNMP data and communicates with the SNMP manager. Typically, the agent is configured to respond only to one or more specific SNMP managers, so that unauthorized parties do not have access to the device. On the Cisco RFGW-10, the Cisco IOS or IOS-XE software runs the SNMP agent software, but it does not become active until it is enabled using the command-line interface (CLI).
- Management Information Base (MIB)—Objects that can be managed by SNMP are defined in MIBs, which are ASCII text files in a structured format. MIBs that are standardized for use industry-wide among multiple vendors are created and maintained by organizations such as the Internet Engineering Task Force (IETF). Vendors, such as Cisco, also create vendor-specific MIBs to manage vendor-specific platforms and features. On the Cisco RFGW-10, MIBs are part of the Cisco IOS or IOS-XE software image. Typically, each new software release includes MIBs that are new or have been modified.

The SNMP manager communicates with the SNMP agent in the following ways:

- GET requests—The SNMP manager obtains information from the device by sending GET requests to the agent. The manager can obtain this information one object at a time using single GET requests.

- SET requests—The SNMP manager configures the device by sending SET requests to the agent. The manager can configure one item at a time using single SET requests, or it can configure multiple parameters using a BULK-SET request.
- Notifications—The SNMP agent asynchronously informs the manager that specific events have occurred by using a trap or inform message (depending on the version of SNMP being used). The network administrator configures the agent for the types of traps and informs it should send. These can range from purely informational messages, such as traffic statistics, to important messages that warn of critical situations and errors, such as a card failure.

SNMP Notifications

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

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

When an agent detects an alarm condition, the agent:

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

SNMP notifications are sent as either:

- Traps—Unreliable messages, which do not require receipt acknowledgment from the SNMP manager.
- Informs—Reliable messages, which are stored in memory until the SNMP manager issues a response. Informs use more system resources than traps.

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

When an agent detects an alarm condition, it logs information about the time, type, and severity of the condition and generates a notification message, which it then sends to a designated IP host. SNMP notifications can be sent as either *traps* or *informs*. See the for instructions on how to enable traps on the Cisco RF Gateway-10 router. Use the **snmp-server host** command to specify whether to send SNMP notifications as traps or informs. See Chapter 4, “Monitoring Notifications,” for information about Cisco RF Gateway-10 router notifications.

SNMP Versions

Cisco IOS and IOS-XE softwares support the following versions of SNMP:

- SNMPv1—The Simple Network Management Protocol: A full Internet standard, defined in RFC 1157. Security is based on community strings.
- SNMPv2c—The community-string-based administrative framework for SNMPv2. SNMPv2c is an update of the protocol operations and data types of SNMPv2c (SNMPv2 classic), and uses the community-based security model of SNMPv1. In particular, SNMPv2c adds support for 64-bit counters.
- SNMPv3—Version 3 of SNMP. SNMPv3 uses the following security features to provide secure access to devices:
  - Message integrity—Ensuring that a packet has not been tampered with in transit.

- Authentication—Determining that the message is from a valid source.
- Encryption—Scrambling the contents of a packet to prevent it from being learned by an unauthorized source.

Tip We recommend using SNMPv3 wherever possible because of its superior security features.

SNMPv1 and SNMPv2c

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

SNMPv2c support includes a retrieval mechanism and more detailed error message reporting to management stations. The retrieval mechanism supports the retrieval of tables and large quantities of information, minimizing the number of round-trip transmissions required.

SNMPv2c improved error handling support. SNMPv1 reported all error conditions using a single error code, but SNMPv2c includes a number of expanded error codes that use different error types to distinguish between different kinds of error conditions.

SNMPv2 also reports three different types of exceptions:

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

SNMPv3

SNMPv3 improves security for SNMP communications by using encryption and by defining security models and security levels:

- Encryption—SNMPv3 supports several industry-standard encryption standards, including the Data Encryption Standard (DES).
- Security Model—An authentication strategy for a user and for the group in which the user resides. Different users can be assigned a different security model, depending on the organization's security structure and needs.
- Security Level—Permitted level of security within a security model. SNMPv1 and SNMPv2c used only a two-stage security level: read-only and read-write. SNMPv3 provides a much greater ability to customize the permission levels for different users.

A combination of a security model and a security level determines which security mechanism is employed when handling an SNMP packet.

SNMP Security Models and Levels

Table 1-1 describes the security models and levels provided by the different SNMP versions.

Table 1-1 SNMP Security Models and Levels

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

You must configure the SNMP agent to use the version of SNMP supported by the management station. An agent can communicate with multiple managers; for this reason, you can configure the Cisco IOS and IOS-XE softwares to support communications with one management station using the SNMPv1 protocol, one using the SNMPv2c protocol, and another using SNMPv3.

Note We recommend using SNMPv3 for all SNMP applications, because of its significant security improvements. In addition, SNMPv3 supports 64-bit counters, which are not supported in SNMPv1. If you use SNMPv1, you can not view any objects that are defined as 64-bit counters.

Requests for Comments

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

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

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

Related Information and Useful Links

The following URLs provide access to general information about Cisco MIBs. Use these links to access MIBs for download, and to access related information (such as application notes and OID listings).

- http://tools.cisco.com/ITDIT/MIBS/servlet/index

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

Cisco Technical Support Information and FAQs

The following URLs provide access to SNMP information developed by the Cisco Technical Assistance Center (TAC):

- http://www.cisco.com/en/US/tech/tk648/tk362/tk605/tsd_technology_support_sub-protocol_home.html is the Cisco Technical Support page for SNMP. It provides links to general SNMP information and tips for using SNMP to gather data.
- SNMP: Frequently Asked Questions About MIBs is a list of frequently asked questions (FAQs) about Cisco MIBs.

SNMP Configuration Information

The *Cisco IOS Configuration Fundamentals and Network Management Configuration Guide, Release 12.3* provides information about configuring SNMP support and SNMP commands.

Note This document has reached End of Life. For more information, see the following End-of-Life Announcement at the following URL: http://www.cisco.com/en/US/docs/ios/redirect/eol.html

Configuring SNMP and MIB Support

This chapter describes how to configure Simple Network Management Protocol (SNMP) and Management Information Base (MIB) support for the Cisco RFGW-10. It includes the following sections:

Downloading and Compiling MIBs

The following sections provide information about how to download and compile MIBs for the Cisco RFGW-10:

Considerations for Working with MIBs

While working with MIBs, consider the following:

- Mismatches on datatype definitions might cause compiler errors or warning messages. For example, the OLD-CISCO-CPU-MIB, OLD-CISCO-MEMORY-MIB, and OLD-CISCO-SYSTEM-MIB each define the following OID differently:

```
OLD-CISCO-CPU-MIB.my
     1cpu    OBJECT IDENTIFIER ::= {local 1 }

OLD-CISCO-MEMORY-MIB.my
     1env    OBJECT IDENTIFIER ::= {local 1 }
```

  To eliminate MIB compiler errors or warning messages for mismatched definitions, edit one of the MIB definitions to match the other. Other types of mismatches include:

```
MIB A
Datatype1 ::= INTEGER(0...100)
```

```
Datatype2 ::= INTEGER(1...50)

MIB B
Datatype1 ::= DisplayString
Datatype2 ::= OCTET STRING (SIZE(0...255))
```

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

  SNMPv2-SMI.my
  SNMPv2-TC.my
  SNMPv2-MIB.my
  RFC1213-MIB.my
  IF-MIB.my
  CISCO-SMI.my
  CISCO-PRODUCTS-MIB.my
  CISCO-TC.my

- To see a particular MIB's dependencies, use the "View and Download MIBs" tool, which is part of the SNMP Object Navigator on the Cisco MIB Tools page. The MIB Locator is available at the following URL:

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

- For information about trap definitions, alternative size definitions, and null OIDs, go to the following URL:

  ftp://ftp.cisco.com/pub/mibs/app_notes/mib-compilers

- For listings of OIDs assigned to MIB objects, go to the following URL:

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

Downloading MIBs

To download the MIBs onto your system, if they are not already there, use the following procedure:

Step 1 Go to the Cisco MIB Tools page at the following URL:

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

If the MIB you want to download is not there, try one of the URLs in Step 4.

Step 2 Click the link for the MIB Locator tool.

Step 3 You can use the MIB Locator tool to show a particular MIB or to show all MIBs for a particular platform or for a particular software release.

Step 4 You can also download industry-standard MIBs from the following URLs:

- http://www.ietf.org
- http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml
- http://www.broadband-forum.org/

Compiling MIBs

If you plan to integrate the Cisco RFGW-10 with an SNMP-based management application, you must also compile the MIBs for that platform. Some SNMP managers do this automatically when you place all of the MIBs in a specific location, while others require you to do this manually. For instructions, see the documentation for that particular manager.

Enabling SNMP Support

The following procedure summarizes how to configure the Cisco RFGW-10 for SNMP support. Note that these basic configuration commands are issued for SNMPv2c. For SNMPv3, you must also set up SNMP users and groups.

Step 1 Set up your basic SNMP configuration through the command-line interface (CLI). (For command and setup information, see the list of documents that follows this procedure.)

Step 2 Define SNMP read-only and read-write communities:

```
RFGW-10(config)# snmp-server community Read_Only_Community_Name ro
RFGW-10(config)# snmp-server community Read_Write_Community_Name rw
```

Step 3 Configure SNMP views (to limit the range of objects accessible to different SNMP user groups):

```
RFGW-10(config)# snmp-server view view_name oid-tree {included | excluded}
```

Step 4 If you plan to enable a number of traps, especially if you plan to enable traps for SYSLOG events and alarms, increase the trap queue size from its default of 10:

```
RFGW-10(config)# snmp-server queue-length queue-size
```

The value of *queue-size* can range from 1 to 1000 traps. We recommend a size of at least 100 for systems that are sending traps for SYSLOG events. The default is 10.

Step 5 See the "Enabling Notifications" section on page 4-2 for information on how to enable traps.

For detailed information about SNMP commands, see the following Cisco documents:

- "System Management" section of the *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.2

Note This document has reached End of Life. For more information, see the following End-of-Life Announcement at the following URL: http://www.cisco.com/en/US/docs/ios/redirect/eol.html

- "System Management" section of the *Cisco IOS Configuration Fundamentals Command Reference*, Release 12.2, available at the following URL:

  http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html

Video Specific MIBs

Cisco IOS Release 12.2(44)SQ release defines the two MIBs, SCTE-HMS-MPEG-MIB and SCTE-HMS-QAM-MIB, to extract the video management object model from MPEG technology. These MIBs provide the functionality to manage:

- MPEG characteristics of input and output transport stream, program stream, and elementary stream.
- Video related statistics.
- QAM configurations for video devices.

These MIBs support various video applications such as Video on Demand, Switched Digital Video, and Digital Broadcast.

DOCSIS and M-CMTS Specific MIBs

Cisco IOS Release 12.2(44)SQ supports several MIBs on the Cisco RFGW-10 for DOCSIS and M-CMTS:

- DOCS-IF-MIB
- DOCS-IF-M-CMTS-MIB
- DTI-MIB
- DOCS-CABLE-DEVICE-MIB
- ENTITY-MIB,IF-MIB

Using the Cisco MIB Locator

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

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

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

https://tools.cisco.com/RPF/register/register.do

MIB Specifications

This chapter describes each Management Information Base (MIB) on the Cisco RFGW-10. Each description lists any constraints on how the MIB or its object identifiers (OIDs) are implemented on the Cisco RFGW-10.

Unless noted otherwise, the Cisco RFGW-10 implementation of a MIB follows the standard MIB that has been defined. Any MIB table or object not listed in the tables is implemented as defined in the standard MIB definition.

Note Not all MIBs included in a Cisco IOS or IOS-XE software release are fully supported by the Cisco RFGW-10. Some MIBs are not supported at all. Other MIBs might work, but they have not been tested on the Cisco RFGW-10. In addition, some MIBs are deprecated but cannot be removed from the software. When a MIB is included in the image, this does not necessarily mean it is supported by the Cisco RFGW-10 platform.

Overview of MIB Support

Support for a particular MIB is included as part of the Cisco IOS or IOS-XE software release. Each version of the Cisco IOS or IOS-XE software contains code that responds to SNMP requests for objects that are in the MIBs that are supported in that release for that particular software image.

Each new release of Cisco IOS or IOS-XE software typically changes that support to some extent, usually involving one or more of the following:

- Fixing a caveat or software defect that is preventing the proper use of the MIB
- Updating the software to support the latest version of the MIB or to support optional objects that were not supported previously
- Adding support for new MIBs that are part of a new feature that is being introduced

The fact that a MIB might be included in a Cisco IOS or IOS-XE software release does not imply that the MIB is fully supported on the Cisco RFGW-10. Similarly, the fact that you can access a particular object in a MIB does not imply that the object is fully supported either.

Note As a general rule, deprecated objects and MIBs should not be used, because they have been replaced by other, more functional objects and MIBs. Also, deprecated objects and MIBs can be removed in a future release without notice.

Note The exact MIB support depends on both the Cisco IOS or IOS-XE software image and the Cisco IOS or IOS-XE software release being used.

Table 3-1 lists the MIBs that are included in Cisco IOS Release 12.2(44) SQ for Cisco RFGW-10. Unless otherwise indicated, each MIB is included in all software images for the indicated release. In some cases, MIBs that are included in the software image are not actually supported or are only partially supported.

Table 3-1 Included MIBs on Cisco RFGW-10 in Cisco IOS Release 12.2(44)SQ.

Cisco RFGW-10 in Cisco IOS Release 12.2(44)SQ	Revision
CISCO-BULK-FILE-MIB	
CISCO-CDP-MIB	
CISCO-CLASS-BASED-QOS-MIB	
CISCO-CONFIG-COPY-MIB	
CISCO-CONFIG-MAN-MIB	
CISCO-ENTITY-EXT-MIB	
CISCO-ENTITY-FRU-CONTROL-MIB	
CISCO-ENTITY-SENSOR-MIB	
CISCO-ENVMON-MIB	
CISCO-FLASH-MIB	
CISCO-FTP-CLIENT-MIB	
CISCO-IMAGE-MIB	
CISCO-L2L3-INTERFACE-CONFIG-MIB	
CISCO-LAG-MIB	
CISCO-MAC-NOTIFICATION-MIB	
CISCO-PAGP-MIB	
CISCO-PING-MIB	
CISCO-PORT-SECURITY-MIB	
CISCO-PORT-STORM-CONTROL-MIB	
CISCO-PRIVATE-VLAN-MIB	
CISCO-PROCESS-MIB	
CISCO-RF-MIB	
CISCO-RTTMON-MIB	
CISCO-STP-EXTENSIONS-MIB	
CISCO-SYSLOG-MIB	
CISCO-VLAN-MEMBERSHIP-MIB	
CISCO-VTP-MIB	
DOCS-CABLE-DEVICE-MIB	9908190000Z
DOCS-IF-M-CMTS-MIB	200511160000Z

Table 3-1 Included MIBs on Cisco RFGW-10 in Cisco IOS Release 12.2(44)SQ.

Cisco RFGW-10 in Cisco IOS Release 12.2(44)SQ	Revision
DOCS-IF-MIB	200212200000Z
DTI-MIB	200606280000Z
ENTITY-MIB	9912070000Z
ETHERLIKE-MIB	
IEEE8021-PAE-MIB	
IEEE8023-LAG-MIB	
IF-MIB	9611031355Z
RMON-MIB	
SCTE-HMS-MPEG-MIB	200706051200Z
SCTE-HMS-QAM-MIB	200708231200Z
SMON-MIB	
SNMP-COMMUNITY-MIB	200003060000Z
SNMP-FRAMEWORK-MIB	9901190000Z
SNMP-MPD-MIB	9905041636Z
SNMP-NOTIFICATION-MIB	9808040000Z
SNMP-PROXY-MIB	
SNMP-TARGET-MIB	9808040000Z
SNMP-USM-MIB	9901200000Z
SNMPv2-MIB	9511090000Z
SNMP-VACM-MIB	9901200000Z
TCP-MIB	
UDP-MIB	

Table 3-2 Included MIBs on Cisco RFGW-10 in Cisco IOS Release IOS-XE 3.2.0SQ

Cisco RFGW-10 in Cisco IOS-XE Release IOS-XE 3.2.0SQ	Revision
DOCS-DRF-MIB	200812090000Z

Table 3-3 Included MIBs on Cisco RFGW-10 in Cisco IOS Release IOS-XE 3.3.0SQ

Cisco RFGW-10 in Cisco IOS-XE Release IOS-XE 3.3.0SQ	Revision
SCTE-HMS-MPEG-MIB	200810031700Z

Table 3-4 Included MIBs on Cisco RFGW-10 in Cisco IOS Release IOS-XE 3.3.1SQ

Cisco RFGW-10 in Cisco IOS-XE Release IOS-XE 3.3.1SQ	Revision
CISCO-ENTITY-REDUNDANCY-MIB	200510010000Z

Table 3-5 Included MIBs on Cisco RFGW-10 in Cisco IOS Release IOS-XE 3.4.0SQ

Cisco RFGW-10 in Cisco IOS-XE Release IOS-XE 3.4.0SQ	Revision
SCTE-HMS-MPEG-MIB	200810031700Z

Table 3-6 Included MIBs on Cisco RFGW-10 in Cisco IOS Release IOS-XE 3.4.1SQ

Cisco RFGW-10 in Cisco IOS-XE Release IOS-XE 3.4.1SQ	Revision
SCTE-HMS-QAM-MIB	200807160305Z

MIB Specifications

This section gives a short summary of each MIB, along with the MODULE-IDENTITY and top-level object identifier (OID) that can be used to access the MIB when using an SNMP manager.

CISCO-ENTITY-REDUNDANCY-MIB

The CISCO-ENTITY-REDUNDANCY-MIB supports configuration, control, and monitoring of redundancy protection for various kinds of components on the managed devices.

Effective with Cisco IOS-XE Release 3.3.1SQ, the Cisco RFGW-10 supports the CISCO-ENTITY-REDUNDANCY-MIB. The Cisco RFGW-10 supports this MIB for Cisco DS-384 and Cisco DS-48 line card redundancy.

It is meant to be generic enough to handle basic redundancy control and monitoring for many types of redundant member components and redundancy architectures as long as there is an Entity MIB entPhysicalIndex and entPhysicalVendorType assigned to each member component. It is designed so that the tables can be augmented in other extension MIBS which build upon this MIB by adding additional objects that may be specific to a particular type of redundancy or member component. This MIB can also be used in cases where some types of redundancy groups and members do not require explicit user configuration. One example may be redundant fan assemblies. In those cases, the managed system should internally assign group and member indexes, so that it can provide read-only access to the group and member tables. This allows MIB monitoring for these types of redundant entities.

The CISCO-ENTITY-REDUNDANCY-MIB is implemented for read only objects. The OID for CISCO-ENTITY-REDUNDANCY-MIB is 1.3.6.1.4.1.9.9.498.

Version: 200510010000Z

MIB Objects

The table below shows the CISCO-ENTITY-REDUNDANCY-MIB table and objects:

Table 7 CISCO-ENTITY-REDUNDANCY-MIB Tables and Objects

Object	Description
ceRedunGroupTypesTable	Lists the basic types of redundancy groups supported on the managed device along with additional information about each group type.
ceRedunVendorTypesTable	Lists all entPhysicalVendorTypes allowed as members for a specific ceRedunGroupTypeIndex on the managed device, inclusive for all configurable values for ceRedunType, ceRedunScope, ceRedunArch, and so on. If the ceRedunGroupDefinitionChanged object changes for a particular ceRedunGroupTypeIndex, then this table may have changed and should be read again. **Note** Although a specific ceRedunGroupTypeIndex may allow groups of different entPhysicalVendorTypes, managed devices typically enforce all members within a specific group to have the same entPhysicalVendorType.
ceRedunInternalStatesTable	Allows the managed system to report a read-only list of internal state numbers and the corresponding descriptions which apply for the members of a particular redundancy group type. If the ceRedunGroupDefinitionChanged object changes for a particular ceRedunGroupTypeIndex, then this table may have changed and should be read again.
ceRedunSwitchoverReasonTable	Allows the managed system to report a read-only list of switchover reason indexes and the corresponding descriptions. If the ceRedunGroupDefinitionChanged object changes for a particular ceRedunGroupTypeIndex, then this table may have changed and should be read again.
ceRedunGroupTable	Lists group configuration and status objects for a specific redundancy group. However, the members are configured separately in the ceRedunMbrTable.
ceRedunMbrConfigTable	Lists the group members and generic redundancy objects which are associated with configuring redundancy group members. The switchover granularity should be for one member at a time. If a member is allowed to be an individual port, then switchovers on multi-port line cards would be expected to take place independently for each port on the line card. But if the members are full line cards, then all ports on the line card would be expected to switch at the same time. The Cisco RFGW-10 supports only full line card redundancy.

Table 7 CISCO-ENTITY-REDUNDANCY-MIB Tables and Objects

Object	Description
ceRedunGroupTypesTable	Lists the basic types of redundancy groups supported on the managed device along with additional information about each group type.
ceRedunMbrStatusTable	Lists the redundancy status and other read-only redundancy objects which are associated with redundancy group members. Status associated with member alarm conditions should be reported separately using the CISCO-ENTITY-ALARM-MIB.
ceRedunCommandTable	Allows switchover commands to be sent to members of configured redundancy groups.

ceRedunGroupTypesTable

The table below lists the MIB objects in ceRedunGroupTypesTable:

Table 8 MIB Objects in ceRedunGroupTypesTable

MIB Object	Description
ceRedunGroupTypeIndex	Contains an index assigned for each type of redundancy group supported on a managed system that requires its own table listing entPhysicalVendorTypes allowed as members for its groups.
ceRedunGroupTypeName	Indicates the textual name of the redundancy group type. The value of this object should be the name of the redundancy group type assigned by the local device as it appears for display commands entered in the device console. Examples are port-group, line card-group, fan-group, and so on.
ceRedunGroupCounts	Indicates the current count of redundancy groups for a specific ceRedunGroupTypeIndex. This count indicates the number of rows in the ceRedunGroupTable for a specific ceRedunGroupTypeIndex.
ceRedunNextUnusedGroupIndex	Indicates the next unused group index available for configuring a new redundancy group for this group type. For Cisco RFGW-10, there is no unused group index and this object returns the value 0.
ceRedunMaxMbrsInGroup	Contains the maximum number of primary plus secondary members allowed in a group for a specific ceRedunGroupTypeIndex. If only 1:1 or 1+1 is supported, this should be 2. If the maximum number is unknown or not determinable, the managed system should return 0. The Cisco RFGW-10 supports 1:1 and 1:N redundancy and the value of this object is set to 10.

Table 8 MIB Objects in ceRedunGroupTypesTable

MIB Object	Description
ceRedunUsesGroupName	Contains a boolean object to indicate whether this type of redundancy group uses the ceRedunGroupString object as a group name identifier. If it is reported as 'true', the ceRedunGroupString name must contain no internal spaces. If it is reported as 'false', the ceRedunGroupString object is just used as an optional description for the group rather than as the group name. For Cisco RFGW-10, it is reported as ‘false’.
ceRedunGroupDefinitionChanged	Contains the value of sysUpTime when there was the most recent change to any objects in the ceRedunGroupTypesTable except for ceRedunGroupCounts or ceRedunNextUnusedGroupIndex. The sysUpTime should also reflect changes to either the ceRedunVendorTypesTable, ceRedunInternalStatesTable, or ceRedunSwitchoverReasonTable. Normally these objects are static, but if there was an in service upgrade to the software image of the managed system then the tables may change and should be read again. If there has been no change since the last initialization of the local network management system, this object should contain the value 0. For Cisco RFGW-10, since the write objects are not supported, it returns the value 0.

ceRedunGroupTable

The table below lists the MIB objects in ceRedunGroupTable:

Table 9 MIB Objects in ceRedunGroupTable

MIB Object	Description
ceRedunGroupIndex	Contains the group number assigned to a particular redundancy group. A group consists of one or more primary members which are protected by one or more secondary members.
ceRedunGroupString	Provides details of group string for corresponding group names. If ceRedunUsesGroupName is 'true' for this redundancy group type, this object is a group name identifier and the value of this object has to be specified and should contain no internal spaces when configuring this group entry. If ceRedunUsesGroupName is 'false', the ceRedunGroupString object is just used as an optional description for the group rather than as the group name. In that case it's allowed to have spaces in the string.
ceRedunGroupRedunType	Indicates the intended type of redundancy protection such as 'yCable' or 'aps' for this redundancy group. The line card redundancy type for Cisco RFGW-10 is internal switch. So, the value of this object is set to other(1) for Cisco RFGW-10.
ceRedunGroupScope	Determines the local/remote scope of the redundancy group. This object may not be modified if the associated ceRedunGroupRowStatus object is equal to active(1). For Cisco RFGW-10, the value of this object is set to local(2).
ceRedunGroupArch	The architecture of the redundancy group, such as 1:1 or 1:n, and so on. This object may not be modified if the associated ceRedunGroupRowStatus object is equal to active(1).

Table 9 MIB Objects in ceRedunGroupTable

MIB Object	Description
ceRedunGroupRevert	Indicates the revertive mode of the redundancy group. nonrevertive(1)—The secondary member remains active until another switchable event takes place. revertive(2)—When the condition that caused a switch to the secondary member has been cleared, a switch is made back to the primary member after a configured delay. Switching should normally be revertive for the 1:n and load-sharing architectures. Switching may optionally be revertive with the 1:1 and 1+1 architectures. This object may not be modified if the associated ceRedunGroupRowStatus object is equal to active(1).
ceRedunGroupWaitToRestore	Indicates the Wait To Restore period in seconds. This object is only applicable to groups which are configured as revertive and does not need to be instantiated for groups which are non-revertive. After clearing of a condition that necessitated an automatic switch, the wait to restore period must elapse before reverting. This is intended to avoid rapid switch oscillations. This object may not be modified if the associated ceRedunGroupRowStatus object is equal to active(1).
ceRedunGroupDirection	Applicable only for those types of redundancy such as APS where switchovers can take place independently at near and far ends of a pair of interconnecting links and does not need to be instantiated for other redundancy types. unidirectional(1)—Switchovers are allowed to take place independently at protection equipment at the near and far ends of interconnecting links. bidirectional(2)—When a switchover happens at the near end protection equipment there is some form of signalling which should cause a corresponding switchover at the far end protection equipment. This object may not be modified if the associated ceRedunGroupRowStatus object is equal to active(1).

Table 9 MIB Objects in ceRedunGroupTable

MIB Object	Description
ceRedunGroupStorageType	Indicates the storage type for this conceptual row. By default, the row will not be saved into non-volatile memory unless this object is set to the value nonVolatile. **Note** Conceptual rows having the value 'readOnly' can be used for redundancy groups that cannot be configured and need not allow write-access to any columnar objects in the row.
ceRedunGroupRowStatus	Provides the configuration status of this redundancy group entry. An entry may not exist in the active RowStatus state unless all configurable read-create objects in the entry have an appropriate value. No other read-create objects in this group may be modified if the ceRedunGroupRowStatus object is equal to active(1). When set to 'notInService', changes may be made to configurable read-create objects. Also, associated ceRedunMbrTable objects may be added, deleted and modified. After modifying a conceptual row in this table, the management client must set this object to 'active' in order for the changes to take effect. For Cisco RFGW-10, the value of this object is set to active(1).

ceRedunMbrConfigTable

The table below lists the MIB objects in ceRedunMbrConfigTable:

Table 10 MIB Objects in ceRedunMbrConfigTable

MIB Object	Description
ceRedunMbrNumber	Assigned as a unique member number within a redundancy group. The value 0 always indicates a secondary member. Primary members should have numbers which are higher than secondary members. **Note** This definition of member values, including the use of the value 0 for the secondary member allows compatibility with existing 1:n SONET APS channel numbering. Yet the numbering definition has also been expanded to allow support for the most general m:n redundancy architectures.
ceRedunMbrPhysIndex	Specifies the entity PhysicalIndex which is being configured as a redundancy member. It is the responsibility of the managed device to enforce any restrictions on matching entPhysicalVendorType, slot positions, and so on. among members of the same redundancy group.

Table 10 MIB Objects in ceRedunMbrConfigTable

MIB Object	Description
ceRedunMbrMode	Sets the role to the 'primary' (working) or 'secondary' (protection) within the redundancy group. The designation as 'primary' or 'secondary' is configured and is static. It doesn't change due to a switchover.
ceRedunMbrAddressType	Specifies the type of address used for the ceRedunMbrAddress object. It does not need to be instantiated when the ceRedunGroupScope value is 'remoteSystem' or 'remoteChassis'. For Cisco RFGW-10, the value of this object is set to 0.
ceRedunMbrRemoteAddress	Specifies the remote management address of the shelf or system where the peer member is expected to be configured. It does not need to be instantiated when the ceRedunGroupScope value is 'remoteSystem' or 'remoteChassis'. For Cisco RFGW-10, the value of this object is set to empty string.
ceRedunMbrPriority	Indicates the priority of the member. For 1:n architectures if the secondary member has already become active for a primary member with a lower priority, it can instead take over for a different primary member if that member has higher priority. This field is only applicable if the member is to be included in a group using the 1:n architecture. It is not applicable if the member is to be included in a group using the 1:1 or 1+1 architecture, and is ignored in that case. For Cisco RFGW-10, the value of this object is set to 0.

Table 10 MIB Objects in ceRedunMbrConfigTable

MIB Object	Description
ceRedunMbrStorageType	Indicates the storage type for this conceptual row. By default, the row will not be saved into non-volatile memory unless this object is set to the value nonVolatile. **Note** Conceptual rows having the value 'readOnly' can be used for redundancy groups that aren't configurable and need not allow write-access to any columnar objects in the row. For Cisco RFGW-10, the value of this object is set to ‘readOnly’(5).
ceRedunMbrRowStatus	Indicates the configuration status of this member entry. A row in the ceRedunMbrConfigTable may not be created, deleted, or set to notInService if the associated ceRedunGroupRowStatus object is equal to active. However, if the ceRedunGroupRowStatus object is equal to notInService, a row may be created, deleted or modified. In other words, a member may not be added, deleted or modified if the including group is active. For Cisco RFGW-10, the value of this object is set to active(1).

ceRedunMbrStatusTable

The table below lists the MIB objects in ceRedunMbrStatusTable:

Table 11 MIB Objects in ceRedunMbrStatusTable

MIB Object	Description
ceRedunMbrStatusCurrent	Indicates the current status bit flags for the member. For Cisco RFGW-10, the following status bit flags are supported: • failure(2) • standby(3) • protectionProvided(4)
ceRedunMbrProtectingMbr	Valid only for a secondary member. When the secondary member is active, this value indicates the primary member it has taken over for. When the secondary member is standby, it should return its own member number. Primary members should return their own member number.

Table 11 MIB Objects in ceRedunMbrStatusTable

MIB Object	Description
ceRedunMbrInternalState	Indicates the current internal state index for a member. The corresponding state category and description can be found in the ceRedunInternalStatesTable. It may include any of the initialization or intermediate progression states necessary to reach a stable active or standby state.
ceRedunMbrSwitchoverCounts	For Cisco RFGW-10, this object is not stored and returns the value 0.
ceRedunMbrLastSwitchover	For Cisco RFGW-10, this object is not stored and returns the value 0.
ceRedunMbrSwitchoverReason	For Cisco RFGW-10, this object is not stored and returns the value 0.
ceRedunMbrSwitchoverSeconds	For Cisco RFGW-10, this object is not stored and returns the value 0.

MIB Constraints

The table below lists the CISCO-ENTITY-REDUNDANCY-MIB constraints:

Table 12 CISCO-ENTITY-REDUNDANCY-MIB Constraints

MIB Object	Notes
ceRedunVendorTypesTable	This table is not supported on the Cisco RFGW-10 and returns null.
ceRedunInternalStatesTable	This table is not supported on the Cisco RFGW-10 and returns null.
ceRedunSwitchoverReasonTable	This table is not supported on the Cisco RFGW-10 and returns null.
ceRedunMembers	This table supports the addition, modification and deletion of members in entity redundancy groups. • ceRedunMbrLastChanged—Not supported, returns zero. • ceRedunMbrStatusLastChanged—Not supported, returns zero.

DOCS-CABLE-DEVICE-MIB

The DOCS-CABLE-DEVICE-MIB contains objects to configure and monitor DOCSIS-compliant Cisco RFGW-10 platforms. This MIB was released as RFC 2669.

The MODULE-IDENTITY for the DOCS-CABLE-DEVICE-MIB is docsDev, and its top-level OID is 1.3.6.1.2.1.69 (iso.org.dod.internet.mgmt-mib-2.docsDev).

MIB Constraints

Table 3-13 lists the constrainst on DOCS-CABLE-DEVICE-MIB.

Table 3-13 DOCS-CABLE-DEVICE-MIB Constraints

MIB Object	Notes
docsDevEventTable	When the table is full and a new event occurs, the oldest event is removed and replaced by the new event. The docsDevEvIndex, however, continues incrementing, up to its maximum value of 2,147,483,647.
• docsDevBase	This group of objects is not supported on the Cisco RFGW-10
• docsDevSoftware	This group of objects is not supported on the Cisco RFGW-10.
• docsDevCpeIpMax	
• docsDevServer	This group of objects is not supported on the Cisco RFGW-10.
• docsDevNmAccess	This group of objects is not supported on the Cisco RFGW-10.
• docsDevFilter	This group of objects is not supported on the Cisco RFGW-10.
• docsDevCpe	This group of objects is not supported on the Cisco RFGW-10.

DOCS-DRF-MIB

This MIB module contains the management objects for the management of the Downstream RF Interface specification. Copyright 1999-2008 Cable Television Laboratories, Inc. All rights reserved.

Version: 200812090000Z

Starting Cisco IOS Release 15.0(2)SQ, the DOCS-DRF-MIB is used to configure Downstream External Physical Interface (DEPI) on the Cisco RFGW-10. Querying these tables displays the QAM interfaces and channels for the Cisco RFGW-10 DS-384 line card.

MIB Objects

Table 3-14 lists the tables and objects.

Table 3-14 DOCS-DRF-MIB Tables and Objects

Object	Description
docsDrfDownstreamTable	Contains extensions for the DOCSIS RFI Downstream docsIfDownstreamChannelTable.
docsDrfDownstreamCapabilities Table	Contains the QAM channel capabilities for the Downstream Interface PHY parameters.
docsDrfGroupDependencyTable	This table describes the rules that identify groups of QAM channels with PHY parameters dependencies. A PHY parameter dependency group means that a set to a QAM channel parameter may affect the value of other QAM Channels in the group. This module refers to TSID group as a PHY dependency Group. This table uses the ENTITY-MIB physical component structure to allows the managed system to describe the QAM channels' PHY parameters dependencies. Examples of PHY dependencies could be usage of adjacent frequencies, or QAM channels of RF ports restricted, or same interleaver value, modulation and J.83 Annex value. Additional details and rules describe the PHY parameter dependency is indicated in docsDrfGroupDependencyType.
docsDrfChannelBlockTable	This table configure attributes of block channels and Controls channel Block Tests. A channel block is an ENTITY-MIB containment of PhysicalClass 'module' that represent an RF connector.

Related Features

The DOCS-DRF-MIB object is associated with the *M-CMTS DEPI* feature.

DOCS-IF-M-CMTS-MIB

The DOCS-IF-M-CMTS-MIBprovides functionality to configure and manage the External PHY interface(DEPI) of the M-CMTS Architecture that is the PHY interface of Cisco RFGW-10 Router.

MIB Constraints

The objects of docsIfMCmtsBaseGroup and docsIfMCmtsCoreGroup are not supported. Table 3-15 lists the constraints on DOCS-IF-M-CMTS-MIB:

Table 3-15 DOCS-IF-M-CMTS-MIB Constraints

MIB Object	Notes
docsIfMCmtsEqamDownstreamTable	
• docsIfMCmtsEqamDownstreamDeviceConfigPhyParamLock	Supported values are: • 0x0 • 0xFF
• docsIfMCmtsEqamDownstreamAllocationType	Supports only docsisOnly(1).
• docsIfMCmtsEqamDownstreamAllocationTimeout	Supports only 0 as value.
• docsIfMCmtsEqamDownstreamDRRPAdvertizing	Supports only false as value.
docsIfMCmtsEqamGlobCfgDownBandwidth	Read-only.
docsIfMCmtsEqamGlobCfgDownSymbolRateM	Read-only.
docsIfMCmtsEqamGlobCfgDownTable	Read-write.
docsIfMCmtsEqamGlobCfgDownSymbolRateN	Read-only.
docsIfMCmtsChannelBlockTable	
• docsIfMCmtsChannelBlockCfgNumberChannels	Supported values are: • 1 • 2 • 4
• docsIfMCmtsChannelBlockTestType	Supported value are: • noTest(1) • offOthersNormal(2) • allOff(3) • onOthersOff(4) • cwOnOthersOff(5)

DOCS-IF-MIB

The DOCS-IF-MIB contains objects to configure and monitor the radio frequency (RF) interfaces on DOCSIS-compliant Cisco RFGW-10 platforms. This MIB was released as RFC 2670. This MIB has been updated to draft-ietf-ipcdn-rfmibv2-05.txt.

The MODULE-IDENTITY for the DOCS-IF-MIB is docsIfMIB, and its top-level OID is 1.3.6.1.2.1.10.127 (iso.org.dod.internet.mgmt.mib-2.transmission.docsIfMIB).

MIB Constraints

Table 3-16 lists the constraint on DOCS-IF-MIB:

Table 3-16 DOCS-IF-MIB Constraints

MIB Object	Notes
docsIfBase	Only docsIfDownstreamChannelTable is supported.
docsIfCm	Not supported.
docsIfCmts	Not supported.
docsIfDownChannelWidth	Supported values are: • 8000000 • 6000000.
docsIfDownChannelModulation	Supported values are: • qam64(3) • qam256(4)
docsIfDownChannelInterleave	Supported values are: • taps8Increment16(3) • taps16Increment8(4) • taps32Increment4(5) • taps64Increment2(6) • taps128Increment1(7)

DTI-MIB

The DTI-MIB supports the Cisco RFGW-10 with the following MIB attributes and values:

- dtiProtocolEntityType—Refer to Table 3-17.
- dtiPathTraceabilityTable—Refer to Table 3-18.
- entPhysicalTable—Refer to the following tables:
  - Table 3-19—entPhysicalTable Entries for the Cisco DTI Card
  - Table 3-20—entPhysicalTable Entries for Cisco DTI Ports
  - Table 3-21—entAliasMappingTable Entries for DTI Interfaces

Table 3-17 dtiProtocolEntityType Entries

MIB Attribute	Value	Notes
dtiProtocolEntityType	root(1), server(2), client(3)	
dtiProtocolClientClockType	ituI(1), ituII(2), ituIII(3), st3(4), dtiClock(5)	

Table 3-17 dtiProtocolEntityType Entries

MIB Attribute	Value	Notes
dtiProtocolServerStatusFlag	unknown(0), warmup(1), freerun(2), fastTrackingMode(3), normalMode(4), holdoverMode(5), clientStable(6), testMode(7)	Server status
dtiProtocolClientStatusFlag	unknown(0), warmup(1), freerun(2), fastTrackingMode(3), normalMode(4), holdoverMode(5), bridgingMode(6), testMode(7)	Client status
dtiProtocolServerToDState	Valid(1) or Invalid(2)	Validity of TOD
dtiProtocolServerToDType	Default(1), userTime(2), ntpv4(3), gps(4)	Current TOD source for DTI connection
dtiProtocolServerToDValue	String	Value of TOD in format DDDDD.YYYY/MM/DD.HH:MM:SS.SHH:F.D
dtiProtocolServerCableAdvanceFlag	Valid(1), invalid(2), manual(3)	Cable advance status
dtiProtocolServerCableAdvanceValue	String	Cable advance value
dtiProtocolClientPhaseError	Signed number	Phase error counter
dtiProtocolClientVersion	Unsigned number	Client DTI version
dtiProtocolClientPathTraceability	Unsigned number	DTI tracibeality
dtiProtocolServerClientStableFlag	Valid(1) or Invalid(2)	Client performance stable status of DTI server frame

Table 3-18 dtiPathTraceabilityTable Entries

MIB Attribute	Value	Notes
dtiPathTraceabilityIndex	Unsigned number	index
dtiPathTraceabilityRootServerInetAddrType	unknown(0), ipv4(1), ipv6(2), ipv4z(3), ipv6z(4), dns(16)	Server address type
dtiPathTraceabilityRootServerInetAddr	<IP address>	Server address
dtiPathTraceabilityRootServerOutPhyIdx	Physical index	
dtiPathTraceabilityServerInetAddrType	unknown(0), ipv4(1), ipv6(2), ipv4z(3), ipv6z(4), dns(16)	
dtiPathTraceabilityServerInetAddr	<IP address>	
dtiPathTraceabilityServerOutPhyIdx	Physical index	

Table 3-18 dtiPathTraceabilityTable Entries

MIB Attribute	Value	Notes
dtiPathTraceabilityRootServerProtVersion		DTI version
dtiPathTraceabilityServerProtVersion		DTI protocol version

MIB Constraints

There are no constraints on this MIB.

ENTITY-MIB

The ENTITY-MIB represents physical and logical entities (components) in the Cisco RFGW-10 and allows SNMP management of those entities. This MIB was released as RFC 2737, *Entity MIB (Version 2)*.

The MIB table entPhysicalTable identifies the physical entities in the Cisco RFGW-10. The entPhysicalTable contains a single row for the chassis and a row for each entity in the chassis. A physical entity may contain other entities (for example, a fan-tray bay may contain a fan-tray module, which may contain one or more fans). The physical hierarchy of system components is determined at run time, based on the actual Cisco RFGW-10 configuration.

The ENTITY-MIB shows information only about hardware devices, not virtual devices.

The MODULE-IDENTITY for the ENTITY-MIB is entityMIB, and its top-level OID is 1.3.6.1.2.1.47 (iso.org.dod.internet.mgmt.mib-2.entityMIB).

MIB objects and related constraints introduced in Cisco IOS Release 12.2(44)SQ are included in Table 3-22.

MIB Constraints

Table 3-19 lists the constraints on ENTITY-MIB:

Table 3-19 entPhysicalTable Entries for the DTI Line Card in Cisco IOS Release 12.2(33)SQ

MIB Attribute	Value	Notes
entPhysicalIndex	Physical index	Set by Entity MIB when instance is created
entPhysicalDescr	Timing Clock card (2 DTI ports)	
entPhysicalVendorType	cevRfgwDtcc	Cisco IOS Release 12.2(44)SQ
entPhysicalContainedIn	Corresponding chassis index	
entPhysicalClass	`module`	
entPhysicalParentRelPos	A number denotes where the TCC w/ DTI card locates.	
entPhysicalName	Linecard(slot slot#)	
entPhysicalHardwareRev	Hardware revision	Read from EEPROM

Table 3-19 entPhysicalTable Entries for the DTI Line Card in Cisco IOS Release 12.2(33)SQ

MIB Attribute	Value	Notes
entPhysicalFirmwareRev	Firmware revision	Read from EEPROM
entPhysicalSoftwareRev	Software revision	Read from EEPROM
entPhysicalSerialNum	N/A	Read from EEPROM
entPhysicalMfgName	Cisco	
entPhysicalModelName	RFGW-TCC1	
entPhysicalAlias		Null by default, can be set.
entPhysicalAssetID		Null by default, can be set.
entPhysicalIsFRU	TRUE(1)	

Table 3-20 entPhysicalTable Entries for the DTI Ports in Cisco IOS Release 12.2(44)SQ

MIB Attribute	Value	Notes
entPhysicalIndex	Physical index	Set when the port is enabled.
entPhysicalDescr		
entPhysicalVendorType	cevPortDti	Cisco IOS Release 12.2(44)SQ
entPhysicalContainedIn	DTI card physical index	e.g. Ge containedin SFP module
entPhysicalClass	Port(10)	
entPhysicalParentRelPos	0	A number given when DTI is enabled, increasing from 0.
entPhysicalName		
entPhysicalHardwareRev	N/A	
entPhysicalFirmwareRev	N/A	
entPhysicalSerialNum	N/A	
entPhysicalMfgname	Cisco	
entPhysicalModelname	N/A	
entPhysicalAlias	NA	
entPhysicalAssetID	NA	
entPhysicalIsFRU	False(2)	

Table 3-21 entAliasMappingTable Entries for DTI interfaces

MIB Attribute	Value	Notes
entAliasLogicalIndexOrZero	0	
entAliasMappingIdentifier	ifIndex.<ifIndex#>	

Table 3-22 ENTITY-MIB Constraints Prior to Cisco IOS Release 12.2(44)SQ

MIB Object	Notes
entPhysicalTable	
• entPhysicalFirmwareRev	Not implemented.
• entPhysicalAlias	Not implemented.
• entPhysicalAssetID	Not implemented.
• entPhysicalSoftwareRev	
• entPhysicalHardwareRev	
• entPhysicalModelName	
• entPhysicalSerialNumber	
• entPhysicalHardwareRev	• Supports the Cisco Unique Device Identifier (UDI). • Contains the Version Identifier (VID).
• entPhysicalModelName	• Supports the Cisco Unique Device Identifier (UDI). • Contains the orderable Product Identifier (PID).
• entPhysicalSerialNumber	• Supports the Cisco unique device Identifier (UDI). • Contains the hardware Serial Number (SN). Note that for non-UDI compliant hardware, these fields might contain a NULL string. Currently, the UDI support is not implemented for the standby PRE. For standby PRE, both entPhysicalHardwareRev and entPhysicalSerialNumber will contain NULL string.
• entPhysicalIsFRU	
• entPhysicalDescr	
• entPhysicalContainedIn	This object accurately displays the location of the FastEthernet network management port on the active PRE-1 module.
entLPMappingTable	Not implemented.

Table 3-22 ENTITY-MIB Constraints Prior to Cisco IOS Release 12.2(44)SQ (continued)

MIB Object	Notes
entAliasMappingTable	• On Cisco RFGW-10, this table does not contain entries for the TCC+ cards. • On Cisco RFGW-10with two redundant PRE modules installed, the entPhysicalDesc entries in the entAliasMappingTable contain two sets of apparently identical references for each PRE module to "Forwarding Processor," "Routing Processor," and "Network Management Ethernet" (the Fast Ethernet port on the PRE modules). This occurs because two PRE modules are installed in the chassis, but only one is active. So the SNMP agent points both entries to the same PRE module. • The entAliasMappingTable supports virtual interfaces only in Cisco IOS Release 12.2(44)SQ and later releases. The entries in this table show the logical upstream interface (as defined by ifIndex from the IF-MIB) that is using each physical upstream connector.

The entPhysicalTable and entAliasMappingTable objects are automatically updated whenever a card is removed or inserted into a slot or when you enter a command at the CLI prompt that affects the operation of the card.

ENTITY-MIB UDI Support

The ENTITY-MIB supports the Cisco compliance effort for a Cisco unique device identifier (UDI) standard which is stored in IDPROM.

The Cisco UDI provides a unique identity for every Cisco product. The unique device identifier is comprised of an orderable product identifier (PID), the version identifier (VID), and the hardware Serial Number (SN). The UDI is stored in IDPROM. The PID, VID, and SN must be stored in the entPhysicalTable:

- PID shall be stored in the entPhysicalModelName object
- VID shall be stored in the entPhysicalHardwareRev object
- SN shall be stored in the entPhysicalSerialNum object

Note The Version ID returns NULL for those old or existing cards whose IDPROMs do not have the Version ID field. Therefore, corresponding entPhysicalHardwareRev returns NULL for cards that do not have the Version ID field in IDPROM. See Identifying Cisco Unique Device Identifiers, page A-22 for a complete description of the Cisco UDI feature.

Each product that is capable of MIB support is required to populate ENTITY-MIB v2 or later with PID, VID, and SN. This compliance is also a requirement of the Consistent Network Element Manageability initiative. If the product uses ENTITY-MIB the data in the following fields should be identical.

ENTITY-MIB v2 (RFC-2737) fields to be populated are:

- Entity-MIB.entPhysicalName (Product Name)

- Entity-MIB.entPhysicalDescr (Product Description)
- Entity-MIB.entPhysicalModelName (PID)
- Entity-MIB.entPhysicalHardwareRev (VID)
- Entity-MIB.entPhysicalSerialNumber (SN)

Overview of the ENTITY-MIB

The following are the most important objects in the ENTITY-MIB for the management of physical entities on the Cisco RFGW-10:

- entPhysicalTable—Describes each physical component (entity) in the Cisco RFGW-10. The table contains a row entry for the top-most entity (the chassis) and then for each entity in the chassis. Each entry provides the name and description of the entry, its type and vendor, and a description of the reason how the entity was first entered into the containment tree.
- entPhysicalIndex—Uniquely identifies each entry. This value is guaranteed to be unique across all equipment in this chassis and across all MIBs, allowing you to correlate the data from several MIBs for any particular entity.
- entAliasMappingTable—Maps each physical port's entPhysicalIndex value to the corresponding ifIndex value in the ifTable in the IF-MIB. This provides a quick way of identifying a particular port with a particular interface.

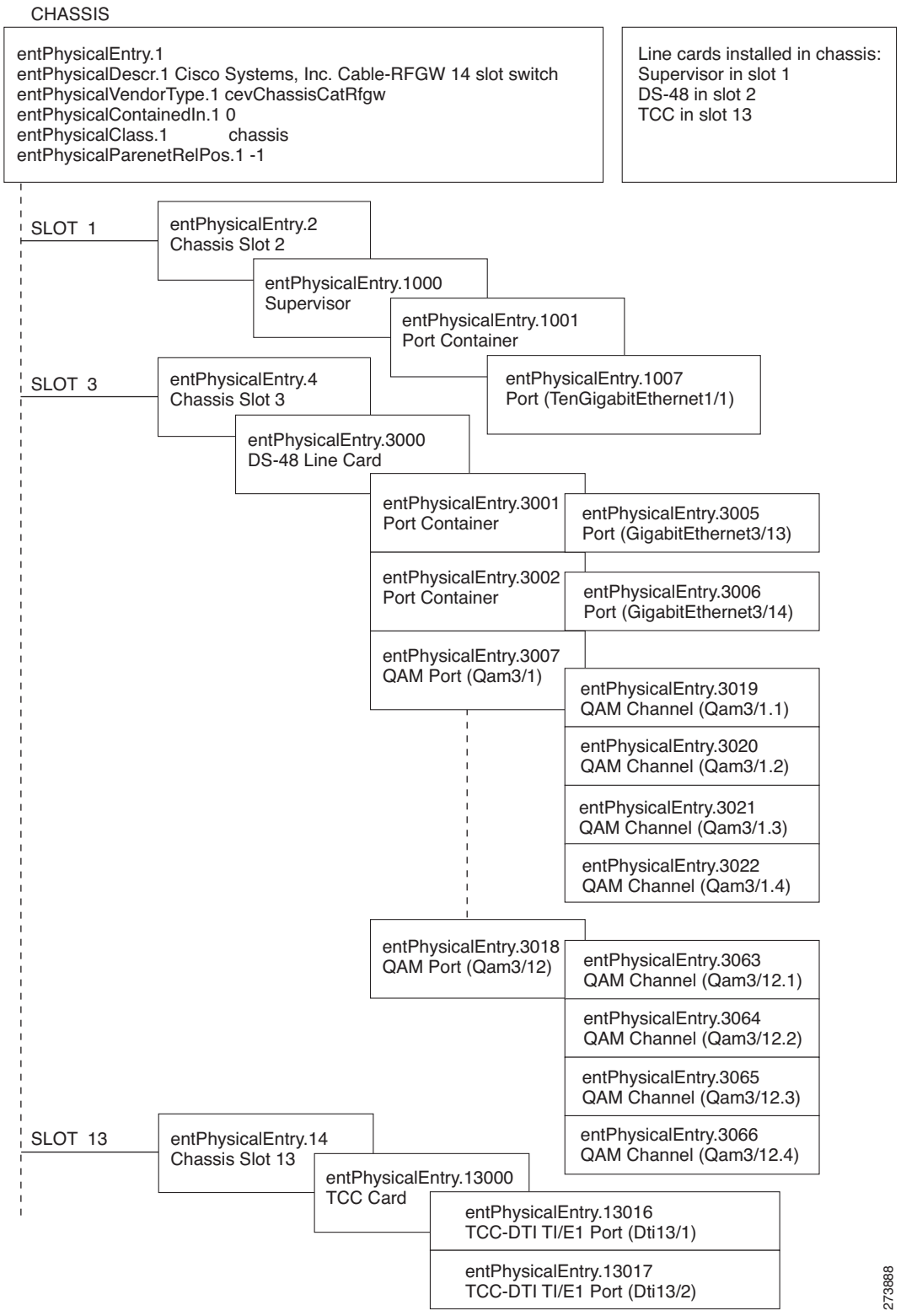
  In Cisco IOS Release 12.2(44)SQ releass, the entAliasMappingTable also shows the mapping of physical upstream connectors to logical upstream interface when virtual interfaces are configured on the Cisco RFGW-10 cable interface line cards. This support also changed the parent-child relationships of cable interface line cards and their ports. For more information, see the "Chassis Slot Layout" section on page 3-25.
- entPhysicalContainsTable—For each physical entity, lists the entPhysicalIndex value for any child objects of the entity. This provides an easy way of creating the container tree for the Cisco RFGW-10, which shows the relationship between physical entities in the chassis.

  Typically, the container tree is organized as follows:

  - The chassis is the top-most level and contains the processor card and chassis slots.
  - Chassis slots contain the individual line cards and I/O controller (if installed).
  - Line cards contain ports (interfaces).

Figure 3-1 shows an overview of the arrangement of objects in a Cisco RFGW-10, with one Fast Ethernet line card and two cable interface line cards.

Figure 3-1 ENTITY-MIB for Cisco RFGW-10 Chassis



Chassis Slot Layout

Table 3-23 provides information about the entities contained in chassis slots on the Cisco RFGW-10.

Table 3-23 Cisco RFGW-10 Chassis Slot Contents

Entity	Can Contain	Notes
Slots 1 to 10	One full-size line card per slot.	For chassis slot containers: • entPhysicalContainedIn is always 1 (which is the entPhysicalIndex of the chassis, which contains all chassis slots). • entPhysicalParentRelPos is different for each chassis slot, to show its position in the chassis. The Cisco RFGW-10 supports only full-size line cards in chassis slots: • entPhysicalContainedIn is different for each line card (because each line card is installed in a different chassis slot). • entPhysicalParentRelPos is always 1 (only one card per slot).

IF-MIB

The IF-MIB describes the attributes of physical and logical interfaces. The Cisco RFGW-10 supports the ifGeneralGroup of MIB objects for all layers (ifIndex, ifDescr, ifType, ifSpeed, ifPhysAddress, ifAdminStatus, ifOperStatus, ifLastChange, ifName, ifLinkUpDownTrapEnable, ifHighSpeed, and ifConnectorPresent). This MIB was released as RFC 2233, *The Interfaces Group MIB Using SMIv2*.

In Cisco IOS Release 12.2(44)SQ release, the Cisco RFGW-10 implemented a cache to allow continuous polling of the ifTable interface counters, without creating spikes in the CPU usage. The cache is updated approximately every 10 seconds, which means that if you read the counter more quickly than that, the SNMP request might not return a new value.

The counters do continue to increment, however, to account for the actual traffic occurring on the interfaces, and another SNMP request in 10 seconds will show the new values. However, the use of the cache means that the counters displayed by the **show interface** command might not exactly match the values returned by the ifTable interface counters.

The MODULE-IDENTITY for the IF-MIB is ifMIB, and its top-level OID is 1.3.6.1.2.1.31 (iso.org.dod.internet.mgmt.mib-2.ifMIB).

IF-MIB Notes

Note It is expected that the interface counters displayed by the **show interface** command might not exactly match the values returned by the ifTable interface counters.

Table 3-24 lists the IF-MIB constraints and notes.

Table 3-24 IF-MIB Constraints

MIB Object	Notes
ifXEntryTable	
• ifAlias	
IfTable	
• ifDescr	The first ifDescr object is typically "Ethernet0/0/0," which is the internal backplane Ethernet interface that the PRE module uses to communicate with the line cards and the secondary PRE module (if installed). This interface cannot be configured or otherwise used, and therefore should be ignored. Note that ifDescr supports Virtual Interfaces only in Cisco IOS Release 12.2(44)SQ and later releases. Also see the ENTITY-MIB for entAliasMappingTable support.
• ifType	Always reports a value of 6 (Ethernet interface) for Ethernet, Fast Ethernet, and Gigabit Ethernet interfaces. This conforms with the recommendations of RFC 2665, and accommodates 10/100/1000 interfaces that negotiate the link speed with the remote end.
• ifInOctets • ifOutOctets	
• ifSpeed	This object has valid values only for the upstream and downstream ifDescr entries. The ifSpeed for the MAC-layer cable interface entry is always 0.
• ifInUcastPkts • ifOutMulticast • ifHCInUcastPkts • ifHCInBroadcastPkts • ifHCOutUcastPkts • ifInUnknownProtos	These objects count both data packets and MAC-layer request packets from cable modems on an upstream, so as to conform to RFC 2670.

SCTE-HMS-MPEG-MIB

This MIB module is for representing MPEG equipment present in the headend and is supported by a SNMP agent. It defines both the MPEG input and output MIB objects for managing MPEG input and output transport streams, programs and elementary streams.

It provides both input and output related statistics, as well as program mapping and video session information. All the tables, except mpegProgramMappingTable and mpegVideoSessionTable, capture and store the information related to active transport streams only. Optional MIB objects have default values defined in this MIB file.

Version: 200810031700Z

MIB Objects

The following table lists the tables and objects.

Table 3-25 SCTE-HMS-MPEG-MIB Tables and Objects

Object	Description
mpegInputTSTable	Specifies the attributes of video sessions or SPTSs.
mpegInputProgTable	Describes the PSI of each incoming program.
mpegProgESTable	Provides information about the elementary streams in a program.
mpegInputStatsTable	Indicates the stream statistics, SPTS or MPTS.
mpegInputUdpOriginationTable	Specifies the UDP unicast or multicast flows of an input transport stream.
mpegInsertPacketTable	Describes the packet insertion information. Typical packets that are inserted at the RF output of a device are PSI, PSIP, and CVCT MPEG packets. These packets have their own PID. This table may be empty if the video device does not support packet insertion or do not have any packet insertion configured.
mpegOutputStatsTable	Specifies the diagnostic statistics objects for the outputs transport stream of an MPEG device.
mpegOutputTSTable	Specifies the attributes of an outgoing transport stream SPTS or MPTS.
mpegOutputProgTable	Describes the PSI of each outgoing program.
mpegOutputProgElemStatsTable	Provides information about the statistics associated with the elementary streams of an MPEG program.
mpegOutputUdpDestinationTable	Specifies the UDP unicast or multicast of the output transport stream this entry references.
mpegProgramMappingTable	Describes the program mappings such as the input destination to the output destination for every program active in the device.
mpegVideoSessionTable	Stores the video session information. The session may be VOD, SDV or DB type. It captures logical information about a video stream, such as source and destination addresses, UDP port, and also ties it with its direct mapping of input and output programs.
mpegVideoSessionPtrTable	Provides a quick reference of the program mapping and input and output transport stream connection information associated with a video Session.
mpegInputTSOutputSessionTable	Specifies the list of output session indexes that the Input transport stream entry.

mpegInputTSTable

The mpegInputTSTable specifies the attributes of an incoming video session or SPTS.

The following table lists the MIB objects in mpegInputTSTable.

Table 3-26 MIB Objects in mpegInputTSTable

MIB Objects	Description
mpegInputTSIndex	Indicates table index.
mpegInputTSType	Indicates the type of stream. Supported streams are: • spts(1) • mpts(2)
mpegInputTSConnectionType	Indicates the type of input flow of the stream. The value udp indicates unicast or multicast udp origination flows.
mpegInputTSConnection	Specifies the start of the data source table that feeds the input stream.
mpegInputTSActiveConnection	Specifies the instance of the data source that feeds the input stream.
mpegInputTSPsiDetected	Indicates whether PSI is detected or not.
mpegInputTSStartTime	Indicates the time MPEG device started receiving the stream (the time the entry was added to the table).
mpegInputTSResourceAllocated	Indicates whether all resources have been allocated for the stream (true) or not (false).
mpegInputTSNumPrograms	Indicates the number of programs in the input transport stream.
mpegInputTSRate	Indicates the data rate of the incoming program or SPTS.
mpegInputTSMaxRate	Indicates the maximum data rate of the incoming VBR stream.
mpegInputTSPatVersion	Indicates the PAT version of the transport stream.
mpegInputTSCatVersion	Not supported. Indicates the CAT version number of transport stream. Set to zero.
mpegInputTSNitPid	Indicates the NIT PID of the input transport stream.
mpegInputTSNumEmms	Indicates the number of EMMs in the input stream. The default value, 9999, indicates that the QAM does not support encryption.
mpegInputTSTSID	Indicates the input TSID of the transport stream. A valid TSID is 16 bits. If TSID is not supported all 32-bits are set to 1.
mpegInputTSLock	Indicates whether the device is locked or not to the physical input. Currently this object is set to notLocked(2). Other supported values are: • locked(1) • intermittent(3) • notIp(4) • notMonitored(5)

mpegInputProgTable

The mpegInputProgTable describes the PSI of each incoming program. Each table entry is identified by ifIndex, mpegInputTSIndex, and mpegInputProgIndex.

The following table lists the MIB objects in mpeginputProgTable:

Table 3-27 MIB Objects in mpegInputProgTable

MIB Objects	Description
mpegInputProgIndex	Specifies the third index of MIB table. The index should uniquely identify a program given a transport stream index.
mpegInputProgNo	Specifies the identifier of the program present in the transport stream of the incoming video stream. This information is contained in the PAT table.
mpegInputProgPmtVersion	Specifies the PMT version of the program.
mpegInputProgPmtPid	Specifies the PMT PID of the program.
mpegInputProgPcrPid	Specifies PCR PID of the program.
mpegInputProgEcmPid	Specifies ECM PID of the program.
mpegInputProgNumElems	Specifies the number of elementary streams in the program. For the current implementation, it is 0.
mpegInputProgNumEcms	Specifies the number of ECMs for the program. The default value, 9999, indicates that the QAM does not support encryption.
mpegInputProgCaDescr	Not supported. Specifies the CA descriptor for the program. If the program does not have an associated CA descriptor, this object has a zero-length string (current implementation). Conditional access descriptor is used to specify system-wide conditional access management information (EMMS) and elementary stream-specific information (ECMs). If any elementary stream is scrambled, a CA descriptor is present for the program containing that elementary stream.
mpegInputProgScte35Descr	Not supported. Specifies the SCTE 35 descriptor. If not present, a zero-length string is returned. For the current implementation, it is a zero-length string.
mpegInputProgScte18Descr	Not supported. Specifies the SCTE 18 descriptor. If not present, a zero-length string is returned. For the current implementation, it is a zero-length string.

mpegProgESTable

The mpegProgESTable contains information about the elementary streams in a program. This table is optional for devices that do not decode and store information such as the video server, encoder, or satellite or if mpegInputTSTransportType is pass-through type for the transport stream.

Each entry in the table is identified by ifIndex, mpegInputTSIndex, mpegInputProgIndex, and mpegProgElemStreamIndex.

The following table lists MIB objects for mpegProgESTable.

Table 3-28 MIB Objects in mpegProgESTable

MIB Objects	Description
mpegProgESIndex	Specifies the Table index.
mpegProgESPID	Specifies the PID for each transport stream packet that carries the program element.
mpegProgESType	Specifies the type of elementary stream (video, audio, or data) of the incoming video session received from the video server. Supported values are: • video(1) • audio(2) • data(3)
mpegProgESCaDescr	Specifies the CA descriptor associated with the elementary stream. If there is no CA descriptor for the elementary stream, this object has a zero-length string.
mpegProgESScte35Descr	Specifies the SCTE 35 descriptor. Returns a zero-length string if it is not supported or is not present. For the current implementation, it is zero-length string.
mpegProgESScte18Descr	Specifies the SCTE 18 descriptor. Returns a zero-length string if it is not supported or is not present. For the current implementation, it is zero-length string.

mpegInputStatsTable

The mpegInputStatsTable contains stream statistics for SPTS or MPTS. Each table entry is identified by ifIndex and mpegInputTSIndex.

The following table lists MIB objects for mpegStatsStreamTable.

Table 3-29 MIB Objects in mpegInputStatsTable

MIB Objects	Description
mpegInputStatsPcrJitter	Specifies the jitter difference between the actual value of the PCR and its expected value in nanoseconds.
mpegInputStatsMaxPacketJitter	Specifies the measurement of the maximum variation in arrival time or delay between the individual packets in milliseconds.
mpegInputStatsPcrPackets	Specifies the number of MPEG transport packets, having PCR, received for the stream.
mpegInputStatsNonPcrPackets	Specifies the number of MPEG transport packets, without PCR, received for the stream.
mpegInputStatsUnexpectedPackets	Specifies the number of transport packets for which PIDs are not expected.

Table 3-29 MIB Objects in mpegInputStatsTable (continued)

MIB Objects	Description
mpegInputStatsContinuityErrors	Specifies the number of continuity counter errors. Transport packets have continuity counters. Successive packets have contiguous values in the continuity counter. Continuity counter error occurs when the successive packets do not have contiguous value and discontinuity indicator is not set.
mpegInputStatsSyncLossPackets	Specifies the number of MPEG packets with missing sync byte per stream. When a sync byte is missing in a MPEG packet this value is incremented.
mpegInputStatsPcrIntervalExceeds	Specifies the number of times that the interval between two successive PCR packets for any program in a transport stream exceeds 100 milliseconds.

mpegInputUdpOriginationTable

The mpegInputUdpOriginationTable specifies the UDP unicast and multicast flows of an input transport stream. For unicast streams, it represents the UDP port and optionally destination IP address of the input TS origination UDP IP flow. For multicast streams, it represents the set of SSM multicast groups of the input TS origination UDP IP flow.

The following table lists the MIB objects for mpegInputUdpOriginationTable.

Table 3-30 MIB Objects in mpegInputUdpOriginationTable

MIB Objects	Description
mpegInputUdpOriginationIndex	Specifies the UDP transport stream origination information.
mpegInputUdpOriginationId	Specifies each UDP IP flow associated with the input transport stream UDP origination.
mpegInputUdpOriginationIfIndex	Specifies the interface index where the UDP flow is received.
mpegInputUdpOriginationInetAddrType	Specifies the address associated with the input transport stream origination UDP IP flow.
mpegInputUdpOriginationScrInetAddr	Specifies the source multicast IP address of the UDP IP flow for multicast transport streams. For unicast UDP IP flows, either the IP source address of the IP flow or all zeros address of known or irrelevant input transport streams are specified.
mpegInputUdpOriginationDestInetAddr	Specifies the group address of the SSM origination input for multicast transport streams. For unicast UDP IP flows, either the IP destination address of the UDP flow or all zeros address for known or irrelevant input transport streams are specified.
mpegInputUdpOriginationDestPort	Specifies the UDP destination port of the UDP IP flow of the input transport stream.

Table 3-30 MIB Objects in mpegInputUdpOriginationTable (continued)

MIB Objects	Description
mpedInputUdpOriginationActive	Specifies the activation state of the UDP flow. The value “true” for a UDP flow data stream indicates it is being sensed or buffered for the input transport stream independently of the flow being used by a video session. For multicast UDP origination flow, the value “true” indicates the UDP flow was successfully joined.
mpegInputUdpOriginationPacketsDetected	Specifies whether the UDP flow packets are being detected.
mpegInputUdpOriginationRank	Specifies the Rank priority used to determine the UDP flow selected for the input transport stream processing and the video session alignments.
mpegInputUdpOriginationInputTSIndex	Specifies the association with the mpegInputTSIndex object.

mpegInsertPacketTable

The mpegInsertPacketTable describes packet insertion information. A packet inserted at the RF output of a QAM is PSI, PSIP, or CVCT MPEG packet. Each packet has a unique PID.

This table is empty if the packet insertion is not supported or packet insertion is not configured on the video device.

The following table lists MIB objects for mpegInsertPacketTable.

Table 3-31 MIB Objects in mpegInsertPacketTable

MIB Objects	Description
mpegInsertPacketIndex	Specifies the index of the list of packets inserted into all MPEG stream.
mpegInsertPacketListId	Specifies the reference number of packets inserted into MPEG stream.
mpegInsertPacketImmediateExecution	Indicates initialization of packet insertion. Always set to true.
mpegInsertPacketStartTime	Specifies Unix epoch start time for insertions (if Immediate Execution is false).
mpegInsertPacketRepeat	Specifies whether the insert packet is one-time or repetitive.
mpegInsertPacketContinuousFlag	Indicates if the packets are sent periodically.
mpegInsertPacketRate	Specifies the insertion rate for the packet list.
mpedInsertPacketDeviceIfIndex	Specifies the IfIndex of the output device interface where the packet is inserted.

mpegOutputStatsTable

The mpegOutputStatsTable specifies the diagnostic statistics objects for the RF outputs of an MPEG device designated by ifIndex and entPhysicalIndex.

The following table lists the MIB objects for mpegOutputStatsTable.

Table 3-32 MIB Objects in mpegOutputStatsTable

MIB Objects	Description
mpegOutputStatsDroppedPackets	Specifies the number of MPEG packets dropped on this output.
mpegOutputStatsFifoOverflow	Specifies the number of FIFO overflows on this output.
mpegOutputStats FifoUnderflow	Specifies the number of FIFO underflows on this output.
mpegOutputStatsDataRate	Specifies the Data rate (bps) for the content on this output.
mpegOutputStatsAvailableBandwidth	Specifies the unused bandwidth on this port.
mpegOutputStatsChannelUtilization	Specifies the current utilization of a channel defined as (measured data rate / total bandwidth) * 100. Returns -1(default) as value if not applicable.
mpegOutputStatsTotalPackets	Specifies the total number of packets sent out for the transport stream for the current the output stream.

mpegOutputTSTable

The mpegoutputTSTable specifies the attributes of an outgoing transport stream SPTS or MPTS.

The following table lists MIB objects for mpegOutputTSTable.

Table 3-33 MIB Objects in mpegOutputTSTable

MIB Objects	Description
mpegOutputTSIndex	Specifies the table index.
mpegOutputTSType	Specifies the type of stream: • spts(1) • mpts(2)
mpegOutputTSConnectionType	Specifies the type of output flow of the stream. • qam indicates a QAM output for the stream. • udp indicates if it is either a unicast or multicast udp destination flow for the stream.
mpegOutputTSConnection	Specifies a reference to the instance of the output connection for the output stream. • Connection type qam—contains the ifIndex of the identifier of the QAM channel of the output stream • Connection type udp—contains the pointer to the UDP destination table used for the output stream.
mpegOutputTSNumPrograms	Specifies the number of programs in the output transport stream.
mpegOutputTSTSID	Specifies the TSID of the output transport stream.
mpegOutputTSNitPid	Not supported. Specifies the NIT PID of the outgoing transport stream. Default value is 65535. **Note** Effective with, Cisco IOS-XE Release 3.4.0SQ, this MIB object is supported on the Cisco RFGW-10.
mpegOutputTSCaPid	Not supported. Specifies the CA PID of the outgoing transport stream. Default value is 65535.
mpegOutputTSCatInsertRate	Not supported. Specifies the CAT insertion rate, expressed in tables/ms. Default value is zero.
mpegOutputTSPatInsertRate	Specifies the PAT insertion rate, expressed in tables/ms.
mpegOutputTSPmtInsertRate	Specifies the PMT insertion rate, expressed in tables/ms.
mpegOutputTSStartTime	Specifies the time the MPEG device started transmitting the output stream.

mpegOutputProgTable

The mpegOutputProgTable describes the PSI of each outgoing program. Each table entity is identified by ifIndex, mpegOutputTSIndex, and mpegOutputProgIndex.

The following table lists MIB objects for mpegOutputProgTable.

Table 3-34 MIB Objects in mpegOutputProgTable

MIB Objects	Description
mpegOutputProgIndex	Specifies the table index for output program.
mpegOutputProgNo	Specifies the identifier of the program present in the transport stream of the outgoing program.
mpegOutputProgPmtVersion	Specifies the PMT version of the outgoing program.
mpegOutputProgPmtPid	Specifies the PMT PID of the outgoing program.
mpegOutputProgPcrPid	Specifies the PCR PID of the outgoing program.
mpegOutputProgEcmPid	Not supported. Specifies the ECM PID of the outgoing program. Default value is 65535.
mpegOutputProgNumElems	Specifies the number of elementary streams in the outgoing program.
mpegOutputProgNumEcms	Specifies the number of ECMs for the outgoing program. The default value, 9999, indicates that the QAM does not support encryption. For the current implementation, the value is 9999.
mpegOutputProgCaDescr	Not supported. Specifies the Conditional Access (CA) descriptor for this program. CA descriptor is used to specify the system-wide conditional access management information (EMMS) and elementary stream-specific information (ECMs). If an elementary stream is scrambled, a CA descriptor for the program contains the elementary stream information. If the outgoing program does not have an associated CA descriptor, this object has a zero-length string. For the current implementation, the value is zero-length string.
mpegOutputProgScte35Descr	Not supported. Specifies the SCTE 35 descriptor in the outgoing program. Returns a zero-length string if it is not supported/not present. For the current implementation, it is zero-length string.
mpegOutputProgScte18Descr	Not supported. Specifies the SCTE 18 descriptor. Returns a zero-length string if it is not supported/not present. For the current implementation, it is zero-length string.

mpegOutputProgElemStatsTable

The mpegOutputProgElemStatsTable contains information associated with the elementary streams of an MPEG program.

This table is optional if mpegInputTSTransportType is pass-through type for the transport stream.

Each table entity is identified by ifIndex, mpegOutputTSIndex, mpegOutputProgIndex, and mpegOutputProgElemStatsIndex.

The following table lists the MIB objects for mpegOutputProgElemStatsTable

Table 3-35 MIB Objects in mpegOutputProgElemStatsTable

MIB Objects	Description
mpegOutputProgElemStatsIndex	Specifies the table index.
mpegOutputProgElemStatsPid	Specifies the PID of output program elementary stream.
mpegOutputProgElemStatsElemType	Specifies the type of elementary stream (video, audio, or data) of the program. For devices, such as QAM, that do not report exact stream type due to scalability issues, this object is set to unknown. Supported values are: • unknown(6) • video(1) • audio(2) • data(3) • scte18(4) • scte35(5)
mpegOutputProgElemStatsDataRate	Specifies the data rate of the elementary stream. Return -1 if not supported. For the current implementation, the value is -1.

mpegOutputUdpDestinationTable

The mpegOutputUdpDestiantionTable specifies the UDP unicast or multicast of the output transport stream this entry references. For unicast it represents the UDP port and optionally, the destination IP address of the output transport stream destination UDP IP flow. For multicast streams, it represents the set of SSM multicast groups of the output transport stream destination UDP IP flow.

The following table lists the MIB objects for the mpegOutputUdpDestiantionTable.

Table 3-36 mpegOutputUdpDestiantionTable

MIB Object	Description
mpegOutputUdpDestinationIndex	Specifies the unique identifier of the UDP output transport stream destination information.
mpegOutputUdpDestinationId	Specifies the unique identifier of each UDP IP flow associated with the output transport stream UDP destination.
mpegOutputUdpDestinationIfIndex	Specifies the interface index where the UDP flow is sent.

Table 3-36 mpegOutputUdpDestiantionTable (continued)

MIB Object	Description
mpegOutputUdpDestinationInetAddrType	Specifies the address type associated with the output transport stream destination UDP IP flow.
mpegOutputUdpDestinationSrcInetAddr	Specifies the source specific multicast (SSM) IP address of the UDP IP flow for multicast transport streams. For unicast UDP IP flow, either the IP source address of the IP flow or all zeros address for known or irrelevant destination transport streams are specified.
mpegOutputUdpDestinationDestInetAddr	Specifies the group address of the SSM destination output transport stream for multicast transport streams. For unicast UDP IP flow, either the IP source address of the IP flow or all zeros address for known or irrelevant destination transport streams are specified.
mpegOutputUdpDestinationDestPort	Specifies the UDP port of the UDP IP flow of the output transport stream.
mpegOutputUdpDestinationOutputTSIndex	Specifies the association with mpegOutputTSIndex entry.

mpegProgramMappingTable

The mpegProgramMappingTable describes program mappings between the input destination and the output destination for every active program in the QAM. The objects of each entry should be read-only for a session-based QAM and read-write for a table-based QAM.

Note Cisco RFGW-10 supports up to 32 programs per QAM, and up to 32 PIDs per program.

Note Cisco RFGW-10 supports read-only for a session/table-based QAM.

The following table lists MIB objects for mpegProgramMappingTable.

Table 3-37 MIB Objects in mpegProgramMappingTable

MIB Objects	Description
mpegProgramMappingIndex	Specifies the table index.
mpegProgramMappingOutputProgIndex	Specifies the mpegOutputProgIndex of the corresponding entry in mpegOutputProgTable for a given output program.
mpegProgramMappingOutputTSIndex	Specifies the mpegOutputTSIndex of the corresponding entry in mpegOutputProgTable for a given output program.
mpegProgramMappingInputProgIndex	Specifies the value of mpegInputProgIndex for the corresponding entry in mpegInputProgTable for a given input program.
mpegProgramMappingInputTSIndex	Specifies the value of mpegInputTSIndex of the entry in mpegInputProgTable for a given input program.

mpegVideoSessionTable

The mpegVideoSessionTable stores the video session information. The session type is VOD, SDV or DB type. A session captures logical information such as source and destination addresses, and UDP port for a video stream and also ties this informationwith direct mapping of input and output programs.

Note The mpegVideoSessionTable captures video sessions provisioned by both session-based and table-based methods.

The following table lists MIB objects for mpegVideoSessionTable.

Table 3-38 MIB Objects in mpegVideoSessionTable

MIB Objects	Description
mpegVideoSessionIndex	Specifies the table index.
mpegVideoSessionPhyMappingIndex	Specifies the mpegProgramMappingIndex of the corresponding mpegProgrammingMappingEntry which contains physical mapping between the input and output program.
mpegVideoSessionPIDRemap	Specifies if a session has PID remapping. A value “true” indicates session has PID remapping.
mpegVideoSessionMode	Indicates if a transport stream is associated with a multiplexed session or is pass-through. For pass-through type of transport stream, all elementary stream level information is optional.
mpegVideoSessionState	Specifies the state of a session. Supported values are: • active(1): Indicates in-use. • provisioned(2): Indicates not-in-use
mpegVideoSessionProvMethod	Specifies the type of session, it can be QAM table-based or session-based.
mpegVideoSessionEncryptionType	Indicates the encryption algorithm of the session. The value 8 indicates PowerKEY encryption and 7 indicates DVB scrambling.
mpegVideoSessionEncryptionInfo	Indicates a reference to the information associated to the session encryption algorithm.
mpegVideoSessionBitRate	Indicates a session throughput (bps).
mpegVideoSessionSessionID	Specifies the session ID associated with the stream in the case of a session-based QAM. This is not applicable to a table-based QAM. This ID may be used by NMS system to uniquely identify an input program to output program mapping.
mpegVideoSessionSelectedInput	Specifies a reference to the input flow currently used in the video session.
mpegVideoSessionSelectedOutput	Specifies a reference to the output flow currently used in the video session.

mpegVideoSessionPtrTable

The mpegVideoSessionPtrTable provides a quick reference of the program mapping and input and output transport stream connection information associated with a video session.

The following table lists the MIB objects for mpegVideoSessionPtrTable.

Table 3-39 mpegVideoSessionPtrTable

MIB Object	Description
mpegVideoSessionPtrInputProgIndex	Indicates the input program index value of the video session.
mpegVideoSessionPtrInputTSIndex	Indicates the input transport stream index value of the video session.
mpegVideoSessionPtrInputTSConnType	Indicates the input transport stream connection type value of the video session.
mpegVideoSessionPtrInputTSConnection	Indicates the input transport stream connection value of the video session.
mpegVideoSessionPtrOutputProgIndex	Indicates the output program index value of the video session.
mpegVideoSessionPtrOutputTSIndex	Indicates the output transport stream index value of the video session.
mpegVideoSessionPtrOutputConnType	Indicates the output transport stream connection type value of the video session.
mpegVideoSessionPtrOutputTSConnection	Indicates the output transport stream connection value of the video session.
mpegVideoSessionPtrStatus	Indicates the status of the sessions. Only active sessions are reported.

mpegInputTSOutputSessionTable

The mpegInputTSOutputSessionTable specifies the list of output session indexes that the input transport stream entry. For unicast sessions, this table points to just one output session. For multicast sessions, it points to all the output sessions using this internally replicated input transport stream.

The following table lists the MIB objects for mpegInputTSOutputSessionTable.

Table 3-40 mpegInputTSOutputSessionTable

MIB Object	Description
mpegInputTSOutputSessionCreate Time	Indicates the creation time of the session.

MIB Constraints

The following table lists the constraints that the router places on objects in the SCTE-HMS-MPEG-MIB.

Table 3-41 SCTE-HMS-MPEG-MIB Constraints

MIB Object	Notes
mpegInputTSCatVersion	Not supported, always returns 0.
mpegInputTSNitPid	Not supported, always returns 65535. **Note** Effective with, Cisco IOS-XE Release 3.4.0SQ, this MIB object is supported on the Cisco RFGW-10.
mpegInputProgEcmPid	Not supported, always returns 65535.
mpegOutputProgEcmPid	Not supported, always returns 65535.
mpegInputTSNumEmms	Not supported, always returns 9999.
mpegInputProgNumEcms	Not supported, always returns 9999.
mpegOutputProgNumEcms	Not supported, always returns 9999.
mpegInputProgCaDescr	Not supported, always returns zero-length string.
mpegInputProgScte35Descr	Not supported, always returns zero-length string.
mpegInputProgScte18Descr	Not supported, always returns zero-length string.
mpegProgElemStreamDescr	Not supported, always returns zero-length string.
mpegProgESScte35Descr	Not supported, always returns zero-length string.
mpegProgESScte18Descr	Not supported, always returns zero-length string.
mpegOutputProgCaDescr	Not supported, always returns zero-length string.
mpegOutputProgScte35Descr	Not supported, always returns zero-length string.
mpegOutputProgScte18Descr	Not supported, always returns zero-length string.
mpegInsertPacketStartTime	Not supported.
mpegOutputStatsChannelUtilization	Not supported, always returns -1.
mpegOutputTSCatInsertRate	Not supported.
mpegOutputProgElemStatsDataRate	No supported, always returns -1.
mpegInputProgNo	Read-only.
mpegOutputProgNo	Read-only.
mpegOutputProgElemStatsPid	Read-only.

SCTE-HMS-QAM-MIB

The SCTE-HMS-QAM-MIB module is for representing Edge QAM equipment present in the headend (or indoor) and is supported by a SNMP agent. It defines QAM channel related configuration MIB objects associated with physical and logical characteristics of QAM channel.

The SCTE-HMS-QAM-MIB contains three read only tables:

- qamChannelTable
- qamChannelCommonTable
- qamConfigTable

Version: 200807160305Z

qamChannelTable

The qamChannelTable describes the configuration and attributes of each QAM channel designated by ifIndex.

Table 3-42 lists the Configuration and Attributes of QAM Channel.

Table 3-42 Configuration and Attributes of QAM Channels

MIB Objects	Description
qamChannelFrequency	Specifies the center frequency of the QAM channel (Hertz).
qamChannelModulationFormat	Specifies the modulation format of the QAM channel. Supported values are: • qam64(1) • qam128(2) • qam256(3) • qam512(4) • qam1024(5)
qamChannelInterleaverLevel	Specifies the interleaver level for FEC coding. This object is ignored when the AnnexMode has the value 'annexA' or 'annexC'. Supported values are: • level1(1) • level2(2)

Table 3-42 Configuration and Attributes of QAM Channels (continued)

MIB Objects	Description
qamChannelInterleaverMode	Specifies the interleaving depth or operation mode of the interleaver. When the qamChannelInterleaverLevel is set to *level1*, a single interleaving depth is supported(fecI128J1). When the qamChannelInterleaverLevel is set to *level2*, all the other interleaving depths are also supported. This object is ignored when AnnexMode has the value *annexA* or *annexC*. Supported values are: • unknown(1) • other(2) • fecI8J16(3) • fecI16J8(4) • fecI32J4(5) • fecI64J2(6) • fecI128J1(7) • fecI12J17(8) • fecI128J2(9) • fecI128J3(10) • fecI128J4(11) • fecI128J5(12) • fecI128J6(13) • fecI128J7(14) • fecI128J8(15)
qamChannelPower	The output power of the QAM channel. If the QAM is muted, this object is set to 0. The output power is in 0.1 dBmV and the value depends on the number of max carriers.
qamChannelSquelch	Indicates whether the QAM port is muted or not. Supported values are: • unmuted(1) • muted(2)

Table 3-42 Configuration and Attributes of QAM Channels (continued)

MIB Objects	Description
qamChannelContWaveMode	Indicates whether Continuous Wave mode is enabled or not for output. Supported values are: • cwmOff(1) • cwmOn(2)
qamChannelAnnexMode	Specifies the ITU-T standard supported by the QAM channel. Supported values are: • unknown(1) • other(2) • annexA(3) • annexB(4) • annexC(5)

qamChannelCommonTable

The qamChannelCommonTable describes QAM channel output bandwidth and utilization information.

Table 3-43 lists MIB objects and description for qamChannelCommonTable.

Table 3-43 MIB Objects in qamChannelCommonTable

MIB Objects	Description
qamChannelCommonOutputBw	Specifies QAM channel output bandwidth or capacity.
qamChannelCommonUtilization	Specifies the utilization of the QAM channel in 0.1 percentage.

qamConfigTable

The qamConfigTable contains the following parameters:

- IP addresses configuration for the QAM channels.
- Program number range associated with QAM channels.
- UDP port range (optional).

Configuring these parameters is necessary when performing session-based provisioning. A session-based provisioning request must conform to the configurations in this table. The QAM channels within a QAM device may be partitioned to support multiple UDP, QAM, or ProgramNo ranges. You can also use this table to show the reserve UDP ports or program numbers for special purpose. The table entries are identified through qamConfigIndex .

Table 3-44 lists MIB objects for qamConfigTable.

Table 3-44 MIB Objects in qamConfigTable

MIB Objects	Description
qamConfigIndex	Specifies table index.
EntPhyIndex	Contains entPhyIndex of RF linecard or 0 if there are no RF linecards. QAM channels are global to the QAM device. An RF linecard contains a number of RF/QAM channels.
qamConfigQamChannelIdMin	Specifies the minimum carrier ID among all the QAM channels associated with the corresponding QAM partition. QAM channel ID maybe within a line card or global depending on EntPhyIndex.
qamConfigQamChannelIdMax	Specifies the maximum carrier ID among all the QAM channels associated with the corresponding QAM partition. QAM channel ID maybe within a line card or global depending on EntPhyIndex.
qamConfigIPAddrType	Specifies the type of the program destination address as defined by inetAddressType. The default value is 1 for ipv4(1).
qamConfigIPAddr	Specifies the IP address of the QAM channel.
qamConfigUdpPortRangeMin	Specifies the lowest UDP port of the UDP port range that can be used on the QAM channel.
qamConfigUdpPortRangeMax	Specifies the highest UDP port within the UDP port range that can be used on the QAM channel.
qamConfigOutputProgNoMin	Specifies the lowest MPEG output program number that can be used on the QAM channel. The default value is 1.
qamConfigOutputProgNoMax	Specifies the highest MPEG output program number that can be used on the QAM channel. The value ranges from 1 to 65535. The default value is 65535.

MIB Constraints

There are no constraints for this MIB.

SNMP-COMMUNITY-MIB

The SNMP-COMMUNITY-MIB contains objects to help support coexistence between the different SNMP versions (SNMPv1, SNMPv2c, and SNMPv3). This MIB was released as RFC 2576, *Coexistence Between Version 1, Version 2, and Version 3 of the Internet-Standard Network Management Framework*.

The MODULE-IDENTITY for the SNMP-COMMUNITY-MIB is snmpCommunityMIB, and its top-level OID is 1.3.6.1.6.3.18 (iso.org.dod.internet.snmpv2.snmpModules.snmpCommunityMIB).

MIB Constraints

There are no constraints on this MIB.

SNMP-FRAMEWORK-MIB

The SNMP-FRAMEWORK-MIB contains objects that describe the SNMP management architecture. This MIB was released as RFC 2571, *An Architecture for Describing SNMP Management Frameworks*.

The MODULE-IDENTITY for the SNMP-FRAMEWORK-MIB is snmpFrameworkMIB, and its top-level OID is 1.3.6.1.6.3.10 (iso.org.dod.internet.snmpv2.snmpModules.snmpFrameworkMIB).

MIB Constraints

There are no constraints on this MIB.

SNMP-MPD-MIB

The SNMP-MPD-MIB contains objects from the agent's Message Processing and Dispatching (MPD) server that report on the the total number of packets received by the SNMP engine that were dropped because they referred to an unknown security model, were for an unknown application, or were otherwise invalid. This MIB was released as RFC 2572, *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)*.

The MODULE-IDENTITY for the SNMP-MPD-MIB is snmpMPDMIB, and its top-level OID is 1.3.6.1.6.3.11 (iso.org.dod.internet.snmpv2.snmpModules.snmpMPDMIB).

MIB Constraints

There are no constraints on this MIB.

SNMP-NOTIFICATION-MIB

The SNMP-NOTIFICATION-MIB contains objects to manage SNMP v3 notifications. This MIB was released as RFC 2573, *SNMP Applications*.

The MIB also defines a set of filters that limit the number of notifications generated by a particular entity (snmpNotifyFilterProfileTable and snmpNotifyFilterTable). Objects in the snmpNotifyTable are used to select entities in the SNMP-TARGET-MIB snmpTargetAddrTable and specify the types of SNMP notifications those entities are to receive.

The MODULE-IDENTITY for the SNMP-NOTIFICATION-MIB is snmpNotificationMIB, and its top-level OID is 1.3.6.1.6.3.13 (iso.org.dod.internet.snmpv2.snmpModules.snmpNotificationMIB).

MIB Constraints

There are no constraints on this MIB.

SNMP-TARGET-MIB

The SNMP-TARGET-MIB contains objects to remotely configure the parameters used by an entity to generate SNMP notifications. The MIB defines the addresses of entities to send SNMP notifications to, and contains a list of tag values that are used to filter the notifications sent to these entities (see the SNMP-NOTIFICATION-MIB). This MIB was defined as part of RFC 2573, *SNMP Applications*.

The MODULE-IDENTITY for the SNMP-TARGET-MIB is snmpTargetMIB, and its top-level OID is 1.3.6.1.6.3.12 (iso.org.dod.internet.snmpv2.snmpModules.snmpTargetMIB).

MIB Constraints

There are no constraints on this MIB.

SNMP-USM-MIB

The SNMP-USM-MIB contains objects that describe the SNMP User-Based Security Model. This MIB was released as RFC 2574, *User-Based Security Model (USM) for Version 3 of the Simple Network Management Protocol (SNMPv3)*.

The MODULE-IDENTITY for the SNMP-USM-MIB is snmpUsmMIB, and its top-level OID is 1.3.6.1.6.3.15 (iso.org.dod.internet.snmpv2.snmpModules.snmpUsmMIB).

MIB Constraints

There are no constraints on this MIB.

SNMP-VACM-MIB

The SNMP-VACM-MIB contains objects to manage the View-Based Access Control Model (VACM) for SNMP clients and managers. This MIB was released as RFC 2575, *View-Based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*.

The MODULE-IDENTITY for the SNMP-VACM-MIB is snmpVacmMIB, and its top-level OID is 1.3.6.1.6.3.16 (iso.org.dod.internet.snmpv2.snmpModules.snmpVacmMIB).

MIB Constraints

There are no constraints on this MIB.

SNMPv2-MIB

The SNMPv2-MIB contains objects to manage SNMPv2 entities on the Cisco RFGW-10. This MIB was released as RFC 1907, Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2).

The MODULE-IDENTITY for the SNMPv2-MIB is snmpMIB, and its top-level OIDs are 1.3.6.1.6.3.1 (iso.org.dod.internet.snmpv2.snmpModules.snmpMIB) and 1.3.6.1.2.1.1 (iso.org.dod.internet.mgmt.mib-2.system).

MIB Constraints

There are no constraints on this MIB.

Monitoring Notifications

This chapter describes the Cisco RF Gateway-10 Router notifications supported by the MIB enhancements feature introduced in Cisco IOS Release 12.2(21)BC. SNMP uses notifications to report events on a managed device. The notifications are traps or informs for different events. The RFGW-10 also supports other notifications not listed.

This chapter contains the following sections:

SNMP Notification Overview

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

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

When an agent detects an alarm condition, the agent:

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

SNMP notifications are sent as either:

- Traps—Unreliable messages, which do not require receipt acknowledgement from the SNMP manager.
- Informs—Reliable messages, which are stored in memory until the SNMP manager issues a response. Informs use more system resources than traps.

To use SNMP notifications on your system, you must specify trap recipients. These recipients indicate where Network Registrar notifications are directed. Traps are enabled depending on the command snmp-server enable traps.

Many commands use the word traps in the command syntax. Unless there is an option in the command to select either traps or informs, the keyword **traps** refers to either traps, informs, or both. Use the **snmp-server host** command to specify whether to send SNMP notifications as traps or informs. The types of traps can be specified in both commands.

Note Most notification types are disabled by default. However, some notification types cannot be controlled with the snmp command. For example, some notification types are always enabled and other types are enabled by a different command. The linkUpDown notifications are controlled by the snmp trap link-status command. If you enter this command with no notification-type keywords, the default is to enable all notification types controlled by this command.

Specify the trap types if you don't want all traps to be sent. Then use multiple snmp-server enable traps commands, one for each of the trap types that you used in the snmp host command. The Event Table must have an entry that specifies the action that is to be performed.

For detailed information about notifications and a list of notification types, see:

- The Traps Sent with SNMP-Server Enabled Traps Configured
- Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2

Enabling Notifications

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

Command line interface (CLI)—Specify the recipient of the trap message and specify the types of traps sent. This command also specifies which types of informs are enabled.

- For detailed procedures, go to:
  - http://www.cisco.com/en/US/tech/tk648/tk362/technologies_tech_note09186a008021de3e.shtml
- Performing an SNMP SET operation using the **setany** command— To enable or disable MIB notifications, perform an SNMP **SET** operation on the a specific object.
  - To enable the notifications set the object to true(1)
  - To disable the notifications, set the object to false(2)

Note If you issue the snmp-server enable traps command without a notification-type argument, the RFGW-10 generates traps for all types of events, which might not be desirable. Some MIBs require the user to set additional objects to enable some notifications.

Cisco SNMP Notifications

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

- Text string—The event display

- Brief description—What the event indicates
- Probable cause—What might have caused the notification
- Recommended action—Recommendation as to what should be done when the particular notification occurs

Note In the following tables, where *no action required* is documented, there might be instances where an application, such as trouble ticketing occurs. For detailed information, go to the following URL: http://www.cisco.com/en/US/products/sw/netmgtsw/ps996/tsd_products_support_series_home.html

Functional Notifications

Table 4-1 lists notifications generated for events that might indicate the failure of the Cisco RF Gateway-10 router or conditions that might affect the RFGW-10 functionality.

Table 4-1 Environmental and Functional Notifications

Event	Description	Probable Cause	Recommended Action
cefcModuleStatusChange	Indicates that the status of a module has changed. A management application can use this trap to update the status of a module it manages.	Module has unknown state	Enter the show module command to view error message details. For Syslog messages associated with this event, consult Messages and Recovery Procedures.
		A line card is provisioned for a slot but it is not present in the slot.	Insert a configured line card in the specific slot.
		Module is operational	No action is required.
		Module has failed due to some condition	Enter the show module command to view error message details. For Syslog messages associated with this event, consult Messages and Recovery Procedures.
cefcPowerStatusChange	Indicates that the power status of a field replaceable unit has changed.	FRU is powered off because of an unknown problem.	Enter the show power command to check the actual power usage.For Syslog messages associated with this event, consult Messages and Recovery Procedures
		FRU is powered on	No action is required.
		FRU is administratively off	No action is required.
		FRU is powered off because available system power is insufficient	Enter the show power command to check the actual power usage.

Table 4-1 Environmental and Functional Notifications (continued)

Event	Description	Probable Cause	Recommended Action
cefcFRUInserted	Indicates that a FRU was inserted. The trap indicates the entPhysicalIndex of the slot that the line card was inserted in.	A new field replaceable unit such as a line card, SIP and SPA modules, fan, port, power supply, or redundant power supply was added.	No action is required; but you can enable this trap through the CLI or by setting cefcMIBEnableStatusNotification to true(1).
cefcFRURemoved	Indicates that a FRU was removed and indicates the entPhysicalIndex of the slot from which the line card was removed.	A field replaceable unit such as line cards, SIP and SPA modules, fan, ports, power supply, or redundant power supply was removed.	Replace the field replaceable unit.

Cisco RFGW-10 Line Card Notifications

These notifications indicate the failure of a line card or error conditions on the card that might affect the functionality of all interfaces and connected customers.

Table 4-2 lists ENTITY-MIB notifications generated by Cisco RF Gateway-10 router cards and SPAs.

Table 4-2 Line Card Notifications

Event	Description	Probable Cause	Recommended Action
entConfigChange	An entry for the line card or a shared port adapter is removed from the entPhysicalTable (which causes the value of entLastchangeTime to change).	A line card was removed.	Replace the field replaceable unit.
ceAlarmAsserted	The agent generates this trap when a physical entity asserts an alarm, such as the power entry module 0 failure.	You manually shut down the line card, then you get the line card error or the alarm *Card Stopped Responding OIR* occurs.	Check the entPhysicalDescr type and take the corresponding action. Since, there are many types of asserted alarms.

Table 4-2 Line Card Notifications (continued)

Event	Description	Probable Cause	Recommended Action
ceAlarmCleared	The agent generates this trap when a physical entity clears a previously asserted alarm or when the core or inlet temperature crosses a threshold, such as inlet critical temperature limit.	The agent generates this trap when: • a physical entity clears a previously asserted alarm • a line card is installed in a slot and the alarm *Active Card Removed OIR* is cleared.	

Notes:

* Sensor entities are the physical entities whose entity class must be defined to type entity sensor(8) in the entPhysicalTable.

* Notifications happen only if the particular entity has an entry in entity table.

* If ceAlarmNotifiesEnable is set to 0, it disables ceAlarmAsserted and ceAlarmCleared notifications. Similarly, when ceAlarmSyslogEnable is set to 0, it disables syslog messages corresponding to alarms.

* If ceAlarmHistTableSize is set to 0, it prevents any history from being retained in the ceAlarmHistTable. In addition whenever the ceAlarmHistTableSize is reset (either increased or decreased) the existing log is deleted.

* When a new alarm condition is detected, the carrier alarm LEDs in the individual line cards are currently set by the line card software. The IOS or IOS-XE alarm subsystem does not control the LEDs.

Link Notifications

Table 4-3 lists notifications generated by the RFGW-10 for link-related (interface) events.

Table 4-3 Interface Notifications

Event	Description	Probable Cause	Recommended Action
linkDown	• Indicates that a link is about to enter the Down state, which means it can not transmit or receive traffic. The ifOperStatus object shows the link's current state. Value is down(2). • Indicates that the wideband downstream ports on the SPA are in a down state.	An internal software error might have occurred.	To see if link traps are enabled or disabled on an interface, check ifLinkUpDownTrapEnable (IF-MIB) for the interface. To enable link traps, set ifLinkUpDownTrapEnable to enabled(1). Enable the IETF (RFC 2233) format of link traps by issuing the CLI command **snmp-server trap link ietf**.
linkUp	• Indicates that a link is about to enter the Up state and the ifOperStatus object shows the link's current status. • Indicates that the wideband downstream ports on the SPA are in a up state.	The port manager reactivated a port in the link-down state during a switchover.	To see if link traps are enabled or disabled on an interface, check ifLinkUpDownTrapEnable (IF-MIB) for the interface. To enable link traps, set ifLinkUpDownTrapEnable to enabled(1). Enable the IETF (RFC 2233) format of link traps by issuing the CLI command **snmp-server trap link ietf**.

Configuration Notifications

Table 4-4 lists notifications generated by the RFGW-10 for events related to system configuration.

Table 4-4 RFGW-10 Configuration Notifications

Event	Description	Probable Cause	Recommended Action
ccCopyCompletion • ccCopyServerAddress • ccCopyFileName • ccCopyState • ccCopyTimeStarted • ccCopyTimeCompleted • ccCopyFailCause	A ccCopyCompletion trap is sent when a config-copy request is completed. The ccCopyFailCause is not instantiated, and hence not included in a trap, when the ccCopyState is successful.	Sent when the RFGW-10 finishes copying a configuration file to or from another location.	Enable this trap by setting ccCopyNotificationOnCompletion to true(1).
ciscoConfigManEvent	The current configuration changed.	Sent when the running configuration changes.	No action required.

Redundancy Configuration Notifications

The table below lists the notifications generated by the Cisco RFGW-10 for events related to redundancy configuration (CISCO-ENTITY-REDUNDANCY-MIB):

Table 4-5 Cisco RFGW-10 Redundancy Configuration Notifications

Event	Description	Probable Cause	Recommended Action
ceRedunEventSwitchoverNotifs	The ceRedunEventSwitchoverNotifs contains two objects, ceRedunMbrProtectingMbr and ceRedunMbrStatusCurrent. This notification is sent when the ceRedunMbrProtectingMbr object changes the value of the secondary line card. For more information on the ceRedunMbrProtectingMbr and ceRedunMbrStatusCurrent objects, see CISCO-ENTITY-REDUNDANCY-MIB.	The secondary line card has become active after the primary line card failover for both 1:1 and 1:N redundancy classes.	Verify the cause for primary line card failover.

Using Cisco RFGW-10 MIBs

This chapter describes the objects and MIBs that are needed to use Simple Network Management Protocol (SNMP) requests to perform the following tasks on a Cisco RFGW-10.

Tips and Guidelines

When using SNMP to manage the Cisco RFGW-10, be aware of the following points.

IF-MIB Caching

In this release, the Cisco RFGW-10 implements a cache to allow continuous polling of the ifTable interface counters, without creating spikes in the CPU usage. An SNMP request for these counters returns the values that were last stored in the counter cache memory, instead of returning the current run-time value of these counters. This improves performance, because it means the Cisco IOS or IOS-XE software does not have to poll each line card to obtain these counters when an SNMP request is made.

The ifTable counter cache is updated approximately every 10 seconds, which means that if you read the ifTable interface counters more quickly than every 10 seconds, the SNMP request might not return new values. The run-time counters do continue to increment, however, to account for the actual traffic occurring on the interfaces, and another SNMP request in 10 seconds does show the new values.

SNMP-Based and CLI-Based Counters

The SNMP specifications do not allow most SNMP-based counters to be cleared, except at system initialization. Instead, during normal operations the counters continue incrementing until they reach their maximum value, at which point they wrap around to zero and continue incrementing again.

This behavior requires the following considerations when managing the RFGW-10 using SNMP commands:

- 32-bit counters—A 32-bit counter wraps around to zero after reaching approximately 4.2 billion. On a busy RFGW-10, this means that byte and packet counters could wrap around after only a few days. To ensure that you are maintaining the correct counts for packets and other objects, regularly poll the desired counters and always save the previous values. Subtract the previous value from the current value, and if the difference between the two counters becomes negative, it indicates that the counters have wrapped.

  To accurately total the counters over a period of several weeks or months, you might also need to keep track of the number of times that the counter wraps during this time period. You should poll the counters often enough so that they do not wrap around to zero more than once without being detected.

Tip Some SNMPv3 MIBs are beginning to include 64-bit counters, as well as 32-bit counters, for many of the same objects. If given a choice, use the 64-bit counters, because they typically will not wrap around to zero for months or years, if ever.

- Counting from a specified event or time period—SNMP-based counters begin incrementing from zero when the RFGW-10 is powered on, and continue incrementing until they wrap. To track the number of packets or other objects from a particular event, you must save the value of the counters at the time of the event. Then when you want to obtain a new packet count, compare the current value of the counters with the saved value.
- Comparison with command-line interface (CLI) values—Many **show** commands have a corresponding **clear** command that resets the counters to zero. The **clear** command, however, affects only the counters that are displayed by the CLI, not the SNMP-based counters. In addition, many CLI-based counters automatically reset whenever a certain function, such as resetting an interface, is performed. This means that the counters displayed using CLI commands are not usually the same as the counters displayed by SNMP commands. Be aware of these differences when comparing the CLI-based and SNMP-based counters.

Route Processor Redundancy (RPR) Modules on the Cisco RFGW-10

On a Cisco RFGW-10 running Cisco IOS Release 12.2(44)SQ, SNMP configuration commands and CLI commands are not synchronized to the standby Supervisor. Hence, the active configuration is lost when a switchover occurs and the standby Supervisor is active. When the RFGW-10 switches back to the original Supervisor, the original configuration is restored.

Use CLI commands for critical configurations and save the configuration to the startup-config to ensure that the critical configuration is active during any switchovers.

Obtaining Basic Information About the RFGW-10

Basic information about the Cisco RFGW-10 can be obtained from objects in the following MIBs:

- OLD-CISCO-CHASSIS-MIB, page A-3
- SNMPv2-MIB, page A-3
- ENTITY-MIB, page A-4

OLD-CISCO-CHASSIS-MIB

The following object in the OLD-CISCO-CHASSIS-MIB provides a convenient location to store the chassis serial number for the RFGW-10, so that it can be easily retrieved when calling Cisco Technical Support:

- chassisId—Provides the serial number or ID number for the chassis, as defined by the **snmp-server chassis-id** command, which is typically used to identify the service contract and levels of service that you have purchased from Cisco Technical Support. This object defaults to the empty string, so you must use the **snmp-server chassis-id** command to set the value of this object before you can retrieve it.

```
csh% getmany -v2c 10.10.11.12 public chassisId

chassisId.0 = STRING: ''NWG121902R6''

sysDescr.0 = STRING: Cisco IOS Software, Catalyst 4500 L3 Switch Software
(RFGW-10-ENTSERVICESK9-M), Version 12.2(122_44_SQ_20081124)SQ EARLY DEPLOYMENT
DATECODE BUILD, synced to V122_43_90_SQ
Copyright (c) 1986-2009 by Cisco Systems, Inc.
Compiled Mon 24-Nov-08 13:58 by jdker

sysObjectID.0 = ciscoProducts.ciscoCatRfgw
sysName.0 = RFGW-10
```

SNMPv2-MIB

The following objects in the SNMPv2-MIB provide basic information about the RFGW-10, its software, and other run-time information:

- sysDescr—Provides an overall description of the RFGW-10, including its model number and the version of Cisco IOS or IOS-XE software that it is running. For example:

```
csh% getmany -v2c 10.10.11.12 public sysDescr

sysDescr.0 = Cisco Internetwork Operating System Software
IOS (tm) 10000 Software (RFGW-10-K8P6-M), Released Version 12.2(15)BC1
Copyright (c) 1986-2009 by cisco Systems, Inc.
Compiled Fri 23-Jan-09 23:56 by atifg
```

- sysObjectID—Provides the specific model number, as it is defined in the CISCO-PRODUCTS-MIB. For example:

```
csh% getmany -v2c 10.10.11.12 public sysObjectId

sysObjectID.0 = ciscoProducts.ciscoRFGW-10
```

- sysName—Provides the host name for the RFGW-10, as assigned by the **hostname** command. For example:

```
csh% getmany -v2c 10.10.11.12 public sysName

sysName.0 = RFGW-10
```

- sysUpTime—Provides the time, in hundredths of a second, since the RFGW-10 was last initialized. For example:

```
csh% getmany -v2c 10.10.11.12 public sysUpTime

sysUpTime.0 = 138389875
```

- sysContact—Provides the name, phone number, or other identifying information for the person or department responsible for this RFGW-10, as it was entered using the **snmp-server contact** command. For example:

```
csh% getmany -v2c 10.10.11.12 public sysContact

sysContact.0 = IT Support at 408-555-1212 or epage it-support
```

- sysLocation—Provides a description of the RFGW-10's location, as it was entered using the **snmp-server location** command. For example:

```
csh% getmany -v2c 10.10.11.12 public sysLocation
```

ENTITY-MIB

The following objects in the ENTITY-MIB provide basic information about the RFGW-10 hardware:

- entPhysicalDescr—Provides a description of each hardware component in the RFGW-10. For example, the following is a typical description for the Cisco RFGW-10 chassis:

```
csh% getnext -v2c 10.10.11.12 public entPhysicalDescr

entPhysicalDescr.1 = Cisco Systems, Inc. RFGW-10 14 slot switch
```

- entPhysicalHardwareRev—Provides the hardware revision of each component, if present and supported for that particular component. For example:

```
csh% getnext -v2c 10.10.11.12 public entPhysicalHardwareRev

entPhysicalHardwareRev.1 = 1.1
```

- entPhysicalSerialNum—Provides the serial number for each component, if present and supported for that particular component. For example:

```
csh% getnext -v2c 10.10.11.12 public entPhysicalSerialNum

entPhysicalSerialNum.1 = NWG121902R6
```

- entPhysicalModelName—Provides the model name for each component, if present and supported for that particular component. For example:

```
csh% getnext -v2c 10.10.11.12 public entPhysicalModelName

entPhysicalModelName.1 = RFGW-10
```

Note Also see the next section for more information about the ENTITY-MIB and how to use it.

Managing Physical Components

The Cisco RFGW-10 supports a number of MIBs for the management of the physical components. These MIBs provide the following functions:

- Organizes the physical entities in the chassis into a containment tree that describes the relationship of each entity to all other entities
- Monitors and configures the status of field-replaceable units (FRUs)
- Maps physical ports to their respective interfaces

- Provides asset information for asset tagging
- Provides firmware and software information for chassis components

See the following sections for a description of each MIB, as well as instructions on how to use the MIBs to track the components in the RFGW-10:

- ENTITY-MIB, page A-5
- Performing Inventory Management, page A-6

Tip To retrieve the chassis serial number for the RFGW-10, retrieve the chassisId object from the OLD-CISCO-CHASSIS-MIB. This object defaults to the empty string, so you must use the **snmp-server chassis-id** command to set the value of this object before you can retrieve it.

ENTITY-MIB

The Cisco RFGW-10 uses the ENTITY-MIB, which is defined as the standard RFC 2737, to manage its physical components, which are known as entities. An entity could be a card, a port on a card, a major subsystem on a card, a slot in the chassis, a field-replaceable unit (FRU), or any other equipment that is installed in the RFGW-10.

The ENTITY-MIB defines a set of objects that uniquely identify each entity in the RFGW-10, using a hierarchical containment tree that shows how each entity relates to each other. Other MIBs can then use the objects defined by the ENTITY-MIB to provide additional information about each entity.

The following are the most important objects in the ENTITY-MIB for the management of physical entities on the RFGW-10:

- entPhysicalTable—Describes each physical component (entity) in the RFGW-10. The table contains a row entry for the top-most entity (the chassis) and then for each entity in the chassis. Each entry provides the name and description of the entry, its type and vendor, and a description of how the entity fits into the containment tree.
- entPhysicalIndex—Uniquely identifies each entry. This value is guaranteed to be unique across all equipment in this chassis and across all MIBs, allowing you to correlate the data from several MIBs for any particular entity.
- entAliasMappingTable—Maps each physical port's entPhysicalIndex value to the corresponding ifIndex value in the ifTable in the IF-MIB. This provides a quick way of identifying a particular port with a particular interface.
- entPhysicalContainsTable—For each physical entity, lists the entPhysicalIndex value for any child objects of the entity. This provides an easy way of creating the container tree for the RFGW-10, which shows the relationship between physical entities in the chassis.

  Typically, the container tree is organized as follows:

  - The chassis is the topmost level and contains the processor card and chassis slots.
  - Chassis slots contain the individual line cards and I/O controller (if installed).
  - Line cards contain ports (interfaces).
  - RFGW-10 interface line cards contain downstream ports.

Performing Inventory Management

The ENTITY-MIB provides all of the information needed to collect an inventory of the physical components in the RFGW-10. The following procedure illustrates one way this can be done, using a RFGW-10. In this example, the RFGW-10 contains the following cards:

Slot 1: RFGW-10 Supervisor V-10GE, 2x10GE(X2) and 4x1GE(SFP)

Slot 2: RFGW-10 Supervisor V-10GE, 2x10GE(X2) and 4x1GE(SFP)

Slot 3: RFGW-10 Universal Downstram EQAM Card, 12 RF ports, 48 QAMs

Slot 13: RFGW-10 Timing, Communication, and Control Card

Slot 14: RFGW-10 Timing, Communication, and Control Card

To collect and organize the information in the ENTITY-MIB, use the following procedure.

Step 1 Collect the list of physical entities by displaying all of the entPhysicalDescr objects. For example:

```
entPhysicalDescr.1 = Cisco Systems, Inc. Cable-RFGW-10 14 slot switch
entPhysicalDescr.2 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.3 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.4 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.5 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.6 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.7 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.8 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.9 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.10 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.11 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.12 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.13 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.14 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.15 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.16 = Cable-RFGW-10 14 slot switch backplane
entPhysicalDescr.17 = Chassis Temperature Sensor
entPhysicalDescr.18 = Container of Fan Tray
entPhysicalDescr.19 = FanTray
entPhysicalDescr.20 = Container of Container of Power Supply
entPhysicalDescr.21 = Container of Power Supply
entPhysicalDescr.22 = Power Supply ( DC 4500W )
entPhysicalDescr.23 = Power Supply Fan Sensor
entPhysicalDescr.23 = Power Supply Fan Sensor
entPhysicalDescr.24 = Container of Power Supply
entPhysicalDescr.1000 = Supervisor V-10GE with 2 10GE X2 ports, and 4 1000BaseX SFP ports
entPhysicalDescr.1001 = Port Container
entPhysicalDescr.1002 = Port Container
entPhysicalDescr.1003 = Port Container
entPhysicalDescr.1004 = Port Container
entPhysicalDescr.1005 = Port Container
entPhysicalDescr.1006 = Port Container
entPhysicalDescr.1007 = 10Gbase-SR
entPhysicalDescr.1009 = 1000BaseSX
entPhysicalDescr.1010 = 1000BaseSX
entPhysicalDescr.2000 = Supervisor V-10GE with 2 10GE X2 ports, and 4 1000BaseX SFP ports
entPhysicalDescr.2001 = Port Container
entPhysicalDescr.2002 = Port Container
entPhysicalDescr.2003 = Port Container
entPhysicalDescr.2004 = Port Container
entPhysicalDescr.2005 = Port Container
entPhysicalDescr.2006 = Port Container
entPhysicalDescr.2007 = 10Gbase-LR
```

```
entPhysicalDescr.2009 = 1000BaseT
entPhysicalDescr.2010 = 1000BaseT
entPhysicalDescr.3000 = 48 QAM with 2 SFP(1000BaseX), 1 ASI
entPhysicalDescr.3001 = Port Container
entPhysicalDescr.3002 = Port Container
entPhysicalDescr.3005 = 1000BaseSX
entPhysicalDescr.3006 = 1000BaseSX
entPhysicalDescr.3007 = QAM port
entPhysicalDescr.3008 = QAM port
entPhysicalDescr.3009 = QAM port
entPhysicalDescr.3010 = QAM port
entPhysicalDescr.3011 = QAM port
entPhysicalDescr.3012 = QAM port
entPhysicalDescr.3013 = QAM port
entPhysicalDescr.3014 = QAM port
entPhysicalDescr.3015 = QAM port
entPhysicalDescr.3016 = QAM port
entPhysicalDescr.3017 = QAM port
entPhysicalDescr.3018 = QAM port
entPhysicalDescr.3019 = QAM channel
entPhysicalDescr.3020 = QAM channel
entPhysicalDescr.3021 = QAM channel
entPhysicalDescr.3022 = QAM channel
entPhysicalDescr.3023 = QAM channel
entPhysicalDescr.3024 = QAM channel
entPhysicalDescr.3025 = QAM channel
entPhysicalDescr.3026 = QAM channel
entPhysicalDescr.3027 = QAM channel
entPhysicalDescr.3028 = QAM channel
entPhysicalDescr.3029 = QAM channel
entPhysicalDescr.3030 = QAM channel
entPhysicalDescr.3031 = QAM channel
entPhysicalDescr.3032 = QAM channel
entPhysicalDescr.3033 = QAM channel
entPhysicalDescr.3034 = QAM channel
entPhysicalDescr.3035 = QAM channel
entPhysicalDescr.3036 = QAM channel
entPhysicalDescr.3037 = QAM channel
entPhysicalDescr.3038 = QAM channel
entPhysicalDescr.3039 = QAM channel
entPhysicalDescr.3040 = QAM channel
entPhysicalDescr.3041 = QAM channel
entPhysicalDescr.3042 = QAM channel
entPhysicalDescr.3043 = QAM channel
entPhysicalDescr.3044 = QAM channel
entPhysicalDescr.3045 = QAM channel
entPhysicalDescr.3046 = QAM channel
entPhysicalDescr.3047 = QAM channel
entPhysicalDescr.3048 = QAM channel
entPhysicalDescr.3049 = QAM channel
entPhysicalDescr.3050 = QAM channel
entPhysicalDescr.3051 = QAM channel
entPhysicalDescr.3052 = QAM channel
entPhysicalDescr.3053 = QAM channel
entPhysicalDescr.3054 = QAM channel
entPhysicalDescr.3055 = QAM channel
entPhysicalDescr.3056 = QAM channel
entPhysicalDescr.3057 = QAM channel
entPhysicalDescr.3058 = QAM channel
entPhysicalDescr.3059 = QAM channel
entPhysicalDescr.3060 = QAM channel
entPhysicalDescr.3061 = QAM channel
entPhysicalDescr.3062 = QAM channel
entPhysicalDescr.3063 = QAM channel
```

```
entPhysicalDescr.3064 = QAM channel
entPhysicalDescr.3065 = QAM channel
entPhysicalDescr.3066 = QAM channel
entPhysicalDescr.13000 = Timing Clock card (2 DTI ports)
entPhysicalDescr.13002 = Rf Switch Card (12 rf ports)
entPhysicalDescr.13003 = Rf Switch Card (12 rf ports)
entPhysicalDescr.13004 = Rf Switch Card (12 rf ports)
entPhysicalDescr.13005 = Rf Switch Card (12 rf ports)
entPhysicalDescr.13006 = Rf Switch Card (12 rf ports)
entPhysicalDescr.13007 = Rf Switch Card (12 rf ports)
entPhysicalDescr.13008 = Rf Switch Card (12 rf ports)
entPhysicalDescr.13009 = Rf Switch Card (12 rf ports)
entPhysicalDescr.13010 = Rf Switch Card (12 rf ports)
entPhysicalDescr.13011 = Rf Switch Card (12 rf ports)
entPhysicalDescr.13012 = Rf Switch Card (12 rf ports)
entPhysicalDescr.13013 = Rf Switch Card (12 rf ports)
entPhysicalDescr.13016 = TCC-DTI TI/E1 input port
entPhysicalDescr.13017 = TCC-DTI TI/E1 input port
entPhysicalDescr.14000 = Timing Clock card (2 DTI ports)
entPhysicalDescr.14016 = TCC-DTI TI/E1 input port
entPhysicalDescr.14017 = TCC-DTI TI/E1 input port
```

Step 2 Obtain additional information about each entPhysicalDescr object by collecting the entPhysicalVendorType, entPhysicalName, and entPhysicalClass objects. Use the index value to match the objects with their corresponding entPhysicalDescr object. Table A-1 shows typical descriptions for the objects used in this example.

Table A-1 Sample entPhysicalDescr Objects and Descriptions

Index #	entPhysicalDescr[1]	entPhysicalVendorType	entPhysicalName	entPhysicalClass
1	Cisco Systems, Inc. Cable-RFGW-10 14 slot switch	cevChassisCatRfgw	Switch System	chassis(3)
2	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 1	container(5)
3	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 2	container(5)
4	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 3	container(5)
5	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 4	container(5)
6	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 5	container(5)
7	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 6	container(5)
8	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 7	container(5)
9	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 8	container(5)
10	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 9	container(5)
11	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 10	container(5)
12	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 11	container(5)
13	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 12	container(5)
14	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 13	container(5)
15	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 14	container(5)
16	Cable-RFGW-10 14 slot switch backplane	cevBackplaneCatRfgw	Backplane	backplane(4)
17	Chassis Temperature Sensor	cevSensorCat4kTemp	Chassis Temperature Sensor	sensor(8)

Table A-1 Sample entPhysicalDescr Objects and Descriptions (continued)

Index #	entPhysicalDescr[1]	entPhysicalVendorType	entPhysicalName	entPhysicalClass
18	Container of Fan Tray	cevContainerCat4kFanTrayBay	Fan Tray Bay	container(5)
19	FanTray	cevFanCat4kFanTray	Fan	fan(7)
20	Container of Container of Power Supply	cevContainerCat4kContainerPSBay	Container of Power Supply Bay	container(5)
21	Container of Power Supply	cevContainerCat4kPowerSupplyBay	Power Supply Bay 1	container(5)
22	Power Supply (DC 4500W)	cevPowerSupplyDc4500W	Power Supply 1	powerSupply(6)
23	Power Supply Fan Sensor	cevFanCat4kPowerSupplyFan	Power Supply 1 Fan	sensor(8)
24	Container of Power Supply	cevContainerCat4kPowerSupplyBay	Power Supply Bay 2	container(5)
1000	Supervisor V-10GE with 2 10GE X2 ports, and 4 1000BaseX SFP ports	cevCat4kWsx451610GE	Linecard(slot 1)	module(9)
1001	Port Container	cevContainer10GigBasePort	Port Container 1/1	container(5)
1002	Port Container	cevContainer10GigBasePort	Port Container 1/2	container(5)
1003	Port Container	cevContainer10GigBasePort	Port Container 1/3	container(5)
1004	Port Container	cevContainer10GigBasePort	Port Container 1/4	container(5)
1005	Port Container	cevContainer10GigBasePort	Port Container 1/5	container(5)
1006	Port Container	cevContainer10GigBasePort	Port Container 1/6	container(5)
1007	10Gbase-SR	cevMX210GBaseSR	TenGigabitEthernet1/1	port(10)
1009	1000BaseSX	cevMGBIC1000BaseSX	GigabitEthernet1/3	port(10)
1010	1000BaseSX	cevMGBIC1000BaseSX	GigabitEthernet1/4	port(10)
2000	Supervisor V-10GE with 2 10GE X2 ports, and 4 1000BaseX SFP ports	cevCat4kWsx451610GE	Linecard(slot 2)	module(9)
2001	Port Container	cevContainer10GigBasePort	Port Container 2/1	container(5)
2002	Port Container	cevContainer10GigBasePort	Port Container 2/2	container(5)
2003	Port Container	cevContainer10GigBasePort	Port Container 2/3	container(5)

Table A-1 Sample entPhysicalDescr Objects and Descriptions (continued)

Index #	entPhysicalDescr[1]	entPhysicalVendorType	entPhysicalName	entPhysicalClass
2004	Port Container	cevContainer10GigBasePort	Port Container 2/4	container(5)
2005	Port Container	cevContainer10GigBasePort	Port Container 2/5	container(5)
2006	Port Container	cevContainer10GigBasePort	Port Container 2/6	container(5)
2007	10Gbase-LR	cevMGBIC1000BaseSX	TenGigabitEthernet2/1	port(10)
2009	1000BaseT	cevMGBIC1000BaseT	GigabitEthernet2/3	port(10)
2010	1000BaseT	cevMGBIC1000BaseT	GigabitEthernet2/4	port(10)
3000	48 QAM with 2 SFP(1000BaseX), 1 ASI	cevRfgwQam48ABC	Linecard(slot 3)	module(9)
	Port Container	cevContainerSFP	Port Container 3/13	container(5)
3002	Port Container	cevContainerSFP	Port Container 3/14	container(5)
3005	1000BaseSX	cevMGBIC1000BaseSX	GigabitEthernet3/13	port(10)
3006	1000BaseSX	cevMGBIC1000BaseSX	GigabitEthernet3/14	port(10)
3007	QAM port	cevRfgwPort4Qam	Qam3/1	module(9)
3008	QAM port	cevRfgwPort4Qam	Qam3/2	module(9)
3009	QAM port	cevRfgwPort4Qam	Qam3/3	module(9)
3010	QAM port	cevRfgwPort4Qam	Qam3/4	module(9)
3011	QAM port	cevRfgwPort4Qam	Qam3/5	module(9)
3012	QAM port	cevRfgwPort4Qam	Qam3/6	module(9)
3013	QAM port	cevRfgwPort4Qam	Qam3/7	module(9)
3014	QAM port	cevRfgwPort4Qam	Qam3/8	module(9)
3015	QAM port	cevRfgwPort4Qam	Qam3/9	module(9)
3016	QAM port	cevRfgwPort4Qam	Qam3/10	module(9)
3017	QAM port	cevRfgwPort4Qam	Qam3/11	module(9)
3018	QAM port	cevRfgwPort4Qam	Qam3/12	module(9)
3019	QAM channel	cevPortRfDs	Qam3/1.1	port(10)
3020	QAM channel	cevPortRfDs	Qam3/1.2	port(10)
3021	QAM channel	cevPortRfDs	Qam3/1.3	port(10)
3022	QAM channel	cevPortRfDs	Qam3/1.4	port(10)
3023	QAM channel	cevPortRfDs	Qam3/2.1	port(10)
3024	QAM channel	cevPortRfDs	Qam3/2.2	port(10)
3025	QAM channel	cevPortRfDs	Qam3/2.3	port(10)
3026	QAM channel	cevPortRfDs	Qam3/2.4	port(10)

Table A-1 Sample entPhysicalDescr Objects and Descriptions (continued)

Index #	entPhysicalDescr[1]	entPhysicalVendorType	entPhysicalName	entPhysicalClass
3027	QAM channel	cevPortRfDs	Qam3/3.1	port(10)
3028	QAM channel	cevPortRfDs	Qam3/3.2	port(10)
3029	QAM channel	cevPortRfDs	Qam3/3.3	port(10)
3030	QAM channel	cevPortRfDs	Qam3/3.4	port(10)
3031	QAM channel	cevPortRfDs	Qam3/4.1	port(10)
3032	QAM channel	cevPortRfDs	Qam3/4.2	port(10)
3033	QAM channel	cevPortRfDs	Qam3/4.3	port(10)
3034	QAM channel	cevPortRfDs	Qam3/4.4	port(10)
3035	QAM channel	cevPortRfDs	Qam3/5.1	port(10)
3036	QAM channel	cevPortRfDs	Qam3/5.2	port(10)
3037	QAM channel	cevPortRfDs	Qam3/5.3	port(10)
3038	QAM channel	cevPortRfDs	Qam3/5.4	port(10)
3039	QAM channel	cevPortRfDs	Qam3/6.1	port(10)
3040	QAM channel	cevPortRfDs	Qam3/6.2	port(10)
3041	QAM channel	cevPortRfDs	Qam3/6.3	port(10)
3042	QAM channel	cevPortRfDs	Qam3/6.4	port(10)
3043	QAM channel	cevPortRfDs	Qam3/7.1	port(10)
3044	QAM channel	cevPortRfDs	Qam3/7.2	port(10)
3045	QAM channel	cevPortRfDs	Qam3/7.3	port(10)
3046	QAM channel	cevPortRfDs	Qam3/7.4	port(10)
3047	QAM channel	cevPortRfDs	Qam3/8.1	port(10)
3048	QAM channel	cevPortRfDs	Qam3/8.2	port(10)
3049	QAM channel	cevPortRfDs	Qam3/8.3	port(10)
3050	QAM channel	cevPortRfDs	Qam3/8.4	port(10)
3051	QAM channel	cevPortRfDs	Qam3/9.1	port(10)
3052	QAM channel	cevPortRfDs	Qam3/9.2	port(10)
3053	QAM channel	cevPortRfDs	Qam3/9.3	port(10)
3054	QAM channel	cevPortRfDs	Qam3/9.4	port(10)
3055	QAM channel	cevPortRfDs	Qam3/10.1	port(10)
3056	QAM channel	cevPortRfDs	Qam3/10.2	port(10)
3057	QAM channel	cevPortRfDs	Qam3/10.3	port(10)
3058	QAM channel	cevPortRfDs	Qam3/10.4	port(10)
3059	QAM channel	cevPortRfDs	Qam3/11.1	port(10)
3060	QAM channel	cevPortRfDs	Qam3/11.2	port(10)
3061	QAM channel	cevPortRfDs	Qam3/11.3	port(10)

Table A-1 Sample entPhysicalDescr Objects and Descriptions (continued)

Index #	entPhysicalDescr[1]	entPhysicalVendorType	entPhysicalName	entPhysicalClass
3062	QAM channel	cevPortRfDs	Qam3/11.4	port(10)
3063	QAM channel	cevPortRfDs	Qam3/12.1	port(10)
3064	QAM channel	cevPortRfDs	Qam3/12.2	port(10)
3065	QAM channel	cevPortRfDs	Qam3/12.3	port(10)
3066	QAM channel	cevPortRfDs	Qam3/12.4	port(10)
13000	Timing Clock card (2 DTI ports)	cevRfgwDtcc	Linecard(slot 13)	module(9)
13002	Rf Switch Card (12 rf ports)	cevRfgwRfswitch	Rf Switch Card (12 rf ports) card in linecard 13	other(1)
13003	Rf Switch Card (12 rf ports)	cevRfgwRfswitch	Rf Switch Card (12 rf ports) card in linecard 13	other(1)
13004	Rf Switch Card (12 rf ports)	cevRfgwRfswitch	Rf Switch Card (12 rf ports) card in linecard 13	other(1)
13005	Rf Switch Card (12 rf ports)	cevRfgwRfswitch	Rf Switch Card (12 rf ports) card in linecard 13	other(1)
13006	Rf Switch Card (12 rf ports)	cevRfgwRfswitch	Rf Switch Card (12 rf ports) card in linecard 13	other(1)
13007	Rf Switch Card (12 rf ports)	cevRfgwRfswitch	Rf Switch Card (12 rf ports) card in linecard 13	other(1)
13008	Rf Switch Card (12 rf ports)	cevRfgwRfswitch	Rf Switch Card (12 rf ports) card in linecard 13	other(1)
13009	Rf Switch Card (12 rf ports)	cevRfgwRfswitch	Rf Switch Card (12 rf ports) card in linecard 13	other(1)
13010	Rf Switch Card (12 rf ports)	cevRfgwRfswitch	Rf Switch Card (12 rf ports) card in linecard 13	other(1)
13011	Rf Switch Card (12 rf ports)	cevRfgwRfswitch	Rf Switch Card (12 rf ports) card in linecard 13	other(1)
13012	Rf Switch Card (12 rf ports)	cevRfgwRfswitch	Rf Switch Card (12 rf ports) card in linecard 13	other(1)
13013	Rf Switch Card (12 rf ports)	cevRfgwRfswitch	Rf Switch Card (12 rf ports) card in linecard 13	other(1)

Table A-1 Sample entPhysicalDescr Objects and Descriptions (continued)

Index #	entPhysicalDescr[1]	entPhysicalVendorType	entPhysicalName	entPhysicalClass
13016	TCC-DTI TI/E1 input port	cevPortDti	Dti13/1	port(10)
13017	TCC-DTI TI/E1 input port	cevPortDti	Dti13/2	port(10)
14000	Timing Clock card (2 DTI ports)	cevRfgwDtcc	Linecard(slot 14)	module(9)
14016	TCC-DTI TI/E1 input port	cevPortDti	Dti14/1	port(10)
14017	TCC-DTI TI/E1 input port	cevPortDti	Dti14/2	port(10)

1. Interfaces are typically identified by the chipset that is being used for the interface's connectors. On cable interfaces, upstreams are further identified by the chipsets that are providing the PHY-layer connectivity.

Step 3 To create the containment tree for the RFGW-10, collect the EntPhysicalContainedIn object for each entPhysicalDescr object. The value in EntPhysicalContainedIn is the index number for the parent (or "container") for the corresponding entPhysicalDescr device.

Table A-2 shows the parent container for the entPhysicalDescr objects being used in this example:

Table A-2 Relationship of EntPhysicalContainedIn to entPhysicalDescr

#	entPhysicalDescr	entPhysicalContainedIn Value	Parent Container
1	Cisco Systems, Inc. Cable-RFGW-10 14 slot switch	0	
2	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
3	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
4	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
5	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
6	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
7	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
8	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
9	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
10	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
11	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
12	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
13	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
14	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
15	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
16	Cable-RFGW-10 14 slot switch backplane	1	Chassis
17	Chassis Temperature Sensor	1	Chassis
18	Container of Fan Tray		Chassis
19	FanTray	18	Container
20	Container of Container of Power Supply	1	Chassis

Table A-2 Relationship of EntPhysicalContainedIn to entPhysicalDescr (continued)

#	entPhysicalDescr	entPhysical ContainedIn Value	Parent Container
21	Container of Power Supply	20	Container
22	Power Supply (DC 4500W)	21	Container of Power Supply
23	Power Supply Fan Sensor	22	Power Supply
24	Container of Power Supply	20	Container of Container of Power Supply
1000	Supervisor V-10GE with 2 10GE X2 ports, and 4 1000BaseX SFP ports	2	Chassis slot
1001	Port Container	1000	Supervisor
1002	Port Container	1000	Supervisor
1003	Port Container	1000	Supervisor
1004	Port Container	1000	Supervisor
1005	Port Container	1000	Supervisor
1006	Port Container	1000	Supervisor
1007	10Gbase-SR	1001	Port Container
1009	1000BaseSX	1003	Port Container
1010	1000BaseSX	1004	Port Container
2000	Supervisor V-10GE with 2 10GE X2 ports, and 4 1000BaseX SFP ports	3	Chassis slot
2001	Port Container	2000	Supervisor
2002	Port Container	2000	Supervisor
2003	Port Container	2000	Supervisor
2004	Port Container	2000	Supervisor
2005	Port Container	2000	Supervisor
2006	Port Container	2000	Supervisor
2007	10Gbase-LR	2001	Port Container
2009	1000BaseT	2003	Port Container
2010	1000BaseT	2004	Port Container
3000	48 QAM with 2 SFP(1000BaseX), 1 ASI	4	Chassis slot
3001	Port Container	3000	DS-48 Line Card
3002	Port Container	3000	DS-48 Line Card
3005	1000BaseSX	3001	Port Container
3006	1000BaseSX	3002	Port Container

Table A-2 Relationship of EntPhysicalContainedIn to entPhysicalDescr (continued)

#	entPhysicalDescr	entPhysical ContainedIn Value	Parent Container
3007	QAM port	3000	DS-48 Line Card
3008	QAM port	3000	DS-48 Line Card
3009	QAM port	3000	DS-48 Line Card
3010	QAM port	3000	DS-49 Line Card
3011	QAM port	3000	DS-50 Line Card
3012	QAM port	3000	DS-51 Line Card
3013	QAM port	3000	DS-52 Line Card
3014	QAM port	3000	DS-53 Line Card
3015	QAM port	3000	DS-54 Line Card
3016	QAM port	3000	DS-55 Line Card
3017	QAM port	3000	DS-56 Line Card
3018	QAM port	3000	DS-57 Line Card
3019	QAM channel	3007	QAM port
3020	QAM channel	3007	QAM port
3021	QAM channel	3007	QAM port
3022	QAM channel	3007	QAM port
3023	QAM channel	3008	QAM port
3024	QAM channel	3008	QAM port
3025	QAM channel	3008	QAM port
3026	QAM channel	3008	QAM port
3027	QAM channel	3009	QAM port
3028	QAM channel	3009	QAM port
3029	QAM channel	3009	QAM port
3030	QAM channel	3009	QAM port
3031	QAM channel	3010	QAM port
3032	QAM channel	3010	QAM port

Table A-2 Relationship of EntPhysicalContainedIn to entPhysicalDescr (continued)

#	entPhysicalDescr	entPhysical ContainedIn Value	Parent Container
3033	QAM channel	3010	QAM port
3034	QAM channel	3010	QAM port
3035	QAM channel	3011	QAM port
3036	QAM channel	3011	QAM port
3037	QAM channel	3011	QAM port
3038	QAM channel	3011	QAM port
3039	QAM channel	3012	QAM port
3040	QAM channel	3012	QAM port
3041	QAM channel	3012	QAM port
3042	QAM channel	3012	QAM port
3043	QAM channel	3013	QAM port
3044	QAM channel	3013	QAM port
3045	QAM channel	3013	QAM port
3046	QAM channel	3013	QAM port
3047	QAM channel	3014	QAM port
3048	QAM channel	3014	QAM port
3049	QAM channel	3014	QAM port
3050	QAM channel	3014	QAM port
3051	QAM channel	3015	QAM port
3052	QAM channel	3015	QAM port
3053	QAM channel	3015	QAM port
3054	QAM channel	3015	QAM port
3055	QAM channel	3016	QAM port
3056	QAM channel	3016	QAM port
3057	QAM channel	3016	QAM port
3058	QAM channel	3016	QAM port
3059	QAM channel	3017	QAM port
3060	QAM channel	3017	QAM port
3061	QAM channel	3017	QAM port
3062	QAM channel	3017	QAM port
3063	QAM channel	3018	QAM port
3064	QAM channel	3018	QAM port
3065	QAM channel	3018	QAM port
3066	QAM channel	3018	QAM port
13000	Timing Clock card (2 DTI ports)	14	Chassis slot

Table A-2 Relationship of EntPhysicalContainedIn to entPhysicalDescr (continued)

#	entPhysicalDescr	entPhysical ContainedIn Value	Parent Container
13002	Rf Switch Card (12 rf ports)	13000	Timing Clock card
13003	Rf Switch Card (12 rf ports)	13000	Timing Clock card
13004	Rf Switch Card (12 rf ports)	13000	Timing Clock card
13005	Rf Switch Card (12 rf ports)	13000	Timing Clock card
13006	Rf Switch Card (12 rf ports)	13000	Timing Clock card
13007	Rf Switch Card (12 rf ports)	13000	Timing Clock card
13008	Rf Switch Card (12 rf ports)	13000	Timing Clock card
13009	Rf Switch Card (12 rf ports)	13000	Timing Clock card
13010	Rf Switch Card (12 rf ports)	13000	Timing Clock card
13011	Rf Switch Card (12 rf ports)	13000	Timing Clock card
13012	Rf Switch Card (12 rf ports)	13000	Timing Clock card
13013	Rf Switch Card (12 rf ports)	13000	Timing Clock card
13016	TCC-DTI TI/E1 input port	13000	Timing Clock card
13017	TCC-DTI TI/E1 input port	13000	Timing Clock card
14000	Timing Clock card (2 DTI ports)	15	Chassis slot
14016	TCC-DTI TI/E1 input port	14000	Timing Clock card
14017	TCC-DTI TI/E1 input port	14000	Timing Clock card

Step 4 (Optional)If a parent object contains multiple children that are the same type of object, such as a RFGW-10 that contains multiple line card slots (Chassis Slots), use the entPhysicalParentRelPos objects to organize the child objects into their proper order. The entPhysicalParentRelPos objects contain an integer that shows the sequential order of the child objects. This integer typically starts incrementing from 0, so that it matches the actual numbering of the physical objects (slot 0 has an entPhysicalParentRelPos value of 0, slot 1 has an entPhysicalParentRelPos value of 1, and so forth).

Note If entPhysicalParentRelPos contains –1, then the object does not have an identifiable relationship with the other objects.

Table A-3 shows how the entPhysicalDescr objects that refer to chassis slots can be put into their physical order by using their entPhysicalParentRelPos values. For example, entPhysicalDescr.4 has an entPhysicalParentRelPos value of 3, which indicates that this slot is slot 3 in the RFGW-10 chassis.

Table A-3 Using entPhysicalParentRelPos to Order entPhysicalDescr Objects

#	entPhysicalDescr	entPhysicalContainedIn	entPhysicalParentRelPos	Physical Slot #
1	Cisco Systems, Inc. Cable-RFGW-10 14 slot switch	0 = TopLevel	-1	N/A
2	Cable-RFGW-10 14 slot switch chassis slot	1 = Chassis	1	Slot 1
3	Cable-RFGW-10 14 slot switch chassis slot	2 = Chassis	2	Slot 2
4	Cable-RFGW-10 14 slot switch chassis slot	3 = Chassis	3	Slot 3
5	Cable-RFGW-10 14 slot switch chassis slot	4 = Chassis	4	Slot 4
6	Cable-RFGW-10 14 slot switch chassis slot	5 = Chassis	5	Slot 5
7	Cable-RFGW-10 14 slot switch chassis slot	6 = Chassis	6	Slot 6
8	Cable-RFGW-10 14 slot switch chassis slot	7 = Chassis	7	Slot 7
9	Cable-RFGW-10 14 slot switch chassis slot	8 = Chassis	8	Slot 8
10	Cable-RFGW-10 14 slot switch chassis slot	9 = Chassis	9	Slot 9
11	Cable-RFGW-10 14 slot switch chassis slot	10 = Chassis	10	Slot 10
12	Cable-RFGW-10 14 slot switch chassis slot	11 = Chassis	11	Slot 11

Table A-3 Using entPhysicalParentRelPos to Order entPhysicalDescr Objects (continued)

#	entPhysicalDescr	entPhysicalContainedIn	entPhysicalParentRelPos	Physical Slot #
13	Cable-RFGW-10 14 slot switch chassis slot	12 = Chassis	12	Slot 12
14	Cable-RFGW-10 14 slot switch chassis slot	13 = Chassis	13	Slot 13
15	Cable-RFGW-10 14 slot switch chassis slot	14 = Chassis	14	Slot 14

Step 5 (Optional)To map a physical interface to its ifIndex, which is defined in IF-MIB and used in other MIBs to uniquely identify a logical interface, use the entAliasMappingIdentifier object.

For example, the following shows the entAliasMappingIdentifier values for the RFGW-10 used in this example. In this example, entPhysicalDescr.3019 (which Table A-1 identifies as the Qam3/1.1 interface) maps to an ifIndex value of 199.

```
entAliasMappingIdentifier.1007.0 = ifIndex.3
entAliasMappingIdentifier.1009.0 = ifIndex.5
entAliasMappingIdentifier.1010.0 = ifIndex.6
entAliasMappingIdentifier.2007.0 = ifIndex.9
entAliasMappingIdentifier.2009.0 = ifIndex.11
entAliasMappingIdentifier.2010.0 = ifIndex.12
entAliasMappingIdentifier.3005.0 = ifIndex.15
entAliasMappingIdentifier.3006.0 = ifIndex.16
entAliasMappingIdentifier.3019.0 = ifIndex.199
entAliasMappingIdentifier.3020.0 = ifIndex.200
entAliasMappingIdentifier.3021.0 = ifIndex.201
entAliasMappingIdentifier.3022.0 = ifIndex.202
entAliasMappingIdentifier.3023.0 = ifIndex.203
entAliasMappingIdentifier.3024.0 = ifIndex.204
entAliasMappingIdentifier.3025.0 = ifIndex.205
entAliasMappingIdentifier.3026.0 = ifIndex.206
entAliasMappingIdentifier.3027.0 = ifIndex.207
entAliasMappingIdentifier.3028.0 = ifIndex.208
entAliasMappingIdentifier.3029.0 = ifIndex.209
entAliasMappingIdentifier.3030.0 = ifIndex.210
entAliasMappingIdentifier.3031.0 = ifIndex.211
entAliasMappingIdentifier.3032.0 = ifIndex.212
entAliasMappingIdentifier.3033.0 = ifIndex.213
entAliasMappingIdentifier.3034.0 = ifIndex.214
entAliasMappingIdentifier.3035.0 = ifIndex.215
entAliasMappingIdentifier.3036.0 = ifIndex.216
entAliasMappingIdentifier.3037.0 = ifIndex.217
entAliasMappingIdentifier.3038.0 = ifIndex.218
entAliasMappingIdentifier.3039.0 = ifIndex.219
entAliasMappingIdentifier.3040.0 = ifIndex.220
entAliasMappingIdentifier.3041.0 = ifIndex.221
entAliasMappingIdentifier.3042.0 = ifIndex.222
entAliasMappingIdentifier.3043.0 = ifIndex.223
entAliasMappingIdentifier.3044.0 = ifIndex.224
entAliasMappingIdentifier.3045.0 = ifIndex.225
entAliasMappingIdentifier.3046.0 = ifIndex.226
entAliasMappingIdentifier.3047.0 = ifIndex.227
entAliasMappingIdentifier.3048.0 = ifIndex.228
entAliasMappingIdentifier.3049.0 = ifIndex.229
```

```
entAliasMappingIdentifier.3050.0 = ifIndex.230
entAliasMappingIdentifier.3051.0 = ifIndex.231
entAliasMappingIdentifier.3052.0 = ifIndex.232
entAliasMappingIdentifier.3053.0 = ifIndex.233
entAliasMappingIdentifier.3054.0 = ifIndex.234
entAliasMappingIdentifier.3055.0 = ifIndex.235
entAliasMappingIdentifier.3056.0 = ifIndex.236
entAliasMappingIdentifier.3057.0 = ifIndex.237
entAliasMappingIdentifier.3058.0 = ifIndex.238
entAliasMappingIdentifier.3059.0 = ifIndex.239
entAliasMappingIdentifier.3060.0 = ifIndex.240
entAliasMappingIdentifier.3061.0 = ifIndex.241
entAliasMappingIdentifier.3062.0 = ifIndex.242
entAliasMappingIdentifier.3063.0 = ifIndex.243
entAliasMappingIdentifier.3064.0 = ifIndex.244
entAliasMappingIdentifier.3065.0 = ifIndex.245
entAliasMappingIdentifier.3066.0 = ifIndex.246
entAliasMappingIdentifier.13016.0 = ifIndex.189
entAliasMappingIdentifier.13017.0 = ifIndex.190
entAliasMappingIdentifier.14016.0 = ifIndex.191
entAliasMappingIdentifier.14017.0 = ifIndex.192
```

Generating SNMP Traps

This section describes how to configure the Cisco RFGW-10 to generate SNMP traps when certain events or conditions occur on the RFGW-10. To use SNMP commands to configure the RFGW-10 to generate SNMP traps, you must define at least one target host to receive the traps, using the following procedure:

Tip You can also use the command-line interface (CLI) to enable and configure the generation of traps on the RFGW-10. For information on using the CLI, see the "Enabling Notifications" section on page 4-2.

Step 1 Create an entry in the snmpTargetAddrTable, which is defined in SNMP-TARGET-MIB, for each host that is to receive traps. Each entry contains the following objects:

- snmpTargetAddrName—Unique string, up to 32 characters long, that identifies this host.
- snmpTargetAddrTDomain—The TCP/IP transport service to be used when delivering traps to this host, typically snmpUDPDomain.
- snmpTargetAddrTAddress—The transport address for the host, typically a six-octet value that is composed of the host's four-byte IP address followed by the two-byte UDP port number to which the traps should be sent.
- snmpTargetAddrTimeout—Maximum period of time, in hundredths of a second, that the Cisco RFGW-10 waits for a response from the host (if any). The default is 1500 (15 seconds).
- snmpTargetAddrRetryCount—Default number of times that the Cisco RFGW-10 resends a trap if a response is not received within the timeout period. The default value is 3 retries.
- snmpTargetAddrTagList—List of tags (defined below) that should be associated with this particular target host. If a host's tag value matches an snmpNotifyTag value, the host receives the types of notifications that are defined by the corresponding snmpNotifyType.

- snmpTargetAddrParams—Arbitrary string, up to 32 characters long, that identifies an entry in the snmpTargetParamsTable, which defines the parameters to be used in generating traps.
- snmpTargetAddrStorageType—Type of storage to be used for this row entry: volatile(2), nonVolatile(3), permanent(4), or readOnly(5). The default is nonVolatile(4).
- snmpTargetAddrRowStatus—Must be set to createAndGo(4) or createAndWait(5) to create this row entry. This object must be set only after all of the other entries in the row have been set.

Step 2 Create an entry in the snmpTargetParamsTable, which is defined in SNMP-TARGET-MIB, to define the SNMP parameters that the RFGW-10 should use when generating SNMP notifications. Each entry contains the following objects:

- snmpTargetParamsName—Unique string, up to 32 characters long, that defines this particular entry. This string is also used in the snmpTargetAddrParams to define the parameters to be used when sending traps to any particular host.
- snmpTargetParamsMPModel—Version of SNMP to be used in sending this trap: 0=SNMPv1, 1=SNMPv2c, and 3=SNMPv3.
- snmpTargetParamsSecurityModel—Version of SNMP security to be used in sending traps: 0=SNMPv1, 1=SNMPv2c, and 3=SNMPv3.
- snmpTargetParamsSecurityName—String, up to 32 characters long, to be used in identifying the Cisco RFGW-10 when sending traps.
- snmpTargetParamsSecurityLevel—Type of security to be used when sending traps: noAuthNoPriv(1), authNoPriv(2), and authPriv(3).
- snmpTargetParamsStorageType—Type of storage to be used for this row entry: volatile(2), nonVolatile(3), permanent(4), or readOnly(5). The default is nonVolatile(4).
- snmpTargetParamsRowStatus—Must be set to createAndGo(4) or createAndWait(5) to create this row entry. This object must be set only after all of the other entries in the row have been set.

Step 3 Create an entry in the snmpNotifyTable, which is defined in the SNMP-NOTIFICATION-MIB. Each row in this table contains the following objects, which define a set of host targets that are to receive traps:

- snmpNotifyName—Unique string, up to 32 characters, that identifies this particular row entry.
- snmpNotifyTag—Arbitrary string, up to 255 characters, that identifies the set of hosts to receive traps. This tag value is matched against the snmpTargetAddrTagList object to determine which hosts should receive which traps.
- snmpNotifyType—Defines the type of trap to be set: trap(1) or inform(2). The default is trap(1).
- snmpNotifyStorageType—Type of storage to be used for this row entry: volatile(2), nonVolatile(3), permanent(4), or readOnly(5). The default is nonVolatile(4).
- snmpNotifyRowStatus—Must be set to createAndGo(4) or createAndWait(5) to create this row entry. This object must be set only after all of the other entries in the row have been set.

Step 4 Optionally create rows in the snmpNotifyFilterProfileTable and snmpNotifyFilterTable, which are defined in the SNMP-NOTIFICATION-MIB. These tables create notification filters that limit the types of notifications that the RFGW-10 sends to particular hosts.

Step 5 Optionally enable traps and notifications to be sent. Most other MIBs include their own objects of NOTIFICATION-TYPE that enable or disable feature-specific traps. These notification objects also define the varbinds that are sent with each trap, which contain the specific information about the event that occurred.

A number of notifications and traps can also be enabled using CLI commands. Table A-4 lists some of the most common traps, how they can be enabled through the CLI, and the situations that generate these traps.

Table A-4 Common Notifications and Traps

Type of Trap	Configuration Command to Enable	Description
Configuration Changes	`snmp-server enable traps entity`	When ENTITY traps are enabled, the RFGW-10 generates an entConfigChange trap when the information in any of the following tables in the ENTITY-MIB changes: • entPhysicalTable • entAliasMappingTable • entPhysicalContainsTable **Note** The SNMP manager should also regularly poll the entLastChangeTime object to detect whether traps were missed due to throttling or transmission loss.
Alarm is Asserted or Cleared	`snmp-server enable traps alarms`	When ALARM traps are enabled, the RFGW-10 generates a trap whenever an alarm is asserted or cleared for physical entities that are defined in the entPhysicalTable in the ENTITY-MIB.

Identifying Cisco Unique Device Identifiers

In order to use UDI retrieval, the Cisco product in use must be UDI-enabled. A UDI-enabled Cisco product supports five required Entity MIB objects. The five Entity MIB v2 (RFC-2737) objects are as follows:

- entPhysicalName
- entPhysicalDescr
- entPhysicalModelName
- entPhysicalHardwareRev
- entPhysicalSerialNum

Although the show inventory command may be available, using that command on devices that are not UDI-enabled produces no output.

Before using the UDI Retrieval feature, you should understand the following concepts:

Unique Device Identifier Overview-Each identifiable product is an entity, as defined by the Entity MIB (RFC-2737) and its supporting documents. Some entities, such as a chassis, will have subentities like slots. An Ethernet switch might be a member of a superentity such as a stack. Most Cisco entities that are orderable products will leave the factory with an assigned UDI.

The UDI information is printed on a label that is affixed to the physical hardware device, and it is also stored electronically on the device in order to facilitate remote retrieval.

A UDI consists of the following elements:

- Product identifier (PID) The PID is the name by which the product can be ordered; it has been historically called the Product Name or Part Number. This is the identifier that one would use to order an exact replacement part.

- Version identifier (VID) The VID is the version of the product. Whenever a product has been revised, the VID will be incremented. The VID is incremented according to a rigorous process derived from Telcordia GR-209-CORE, an industry guideline that governs product change notices.
- Serial number (SN) The SN is the vendor-unique serialization of the product. Each manufactured product will carry a unique serial number assigned at the factory, which cannot be changed in the field. This is the means by which to identify an individual, specific instance of a product.