Data-Over-Cable Service Interface Specifications
DCA - MHAv2

Remote PHY OSS Interface Specification

CM-SP-R-OSSI-I01-150817

ISSUED

DISCLAIMER

This document is furnished on an "AS IS" basis and neither CableLabs nor its members provides any representation or warranty, express or implied, regarding the accuracy, completeness, noninfringement, or fitness for a particular purpose of this document, or any document referenced herein. Any use or reliance on the information or opinion in this document is at the risk of the user, and CableLabs and its members shall not be liable for any damage or injury incurred by any person arising out of the completeness, accuracy, or utility of any information or opinion contained in the document.

CableLabs reserves the right to revise this document for any reason including, but not limited to, changes in laws, regulations, or standards promulgated by various entities, technology advances, or changes in equipment design, manufacturing techniques, or operating procedures described, or referred to, herein.

This document is not to be construed to suggest that any company modify or change any of its products or procedures, nor does this document represent a commitment by CableLabs or any of its members to purchase any product whether or not it meets the characteristics described in the document. Unless granted in a separate written agreement from CableLabs, nothing contained herein shall be construed to confer any license or right to any intellectual property. This document is not to be construed as an endorsement of any product or company or as the adoption or promulgation of any guidelines, standards, or recommendations.

Document Status Sheet

Document Control Number:	CM-SP-R-OSSI-I01-150817			
Document Title:	Remote PHY OSS Interface Specification			
Revision History:	I01 - Released 08/17/2015			
Date:	August 17, 2015			
Status:	~~Work in Progress~~	~~Draft~~	**Issued**	~~Closed~~
Distribution Restrictions:	~~Author Only~~	~~CL/Member~~	~~CL/ Member/ Vendor~~	**Public**

Key to Document Status Codes

Work in Progress	An incomplete document, designed to guide discussion and generate feedback that may include several alternative requirements for consideration.
Draft	A document in specification format considered largely complete, but lacking review by Members and vendors. Drafts are susceptible to substantial change during the review process.
Issued	A generally public document that has undergone Member and Technology Supplier review, cross-vendor interoperability, and is for Certification testing if applicable. Issued Specifications are subject to the Engineering Change Process.
Closed	A static document, reviewed, tested, validated, and closed to further engineering change requests to the specification through CableLabs.

Trademarks

Contents

Figures

Tables

This page left blank intentionally.

1 SCOPE

1.1 Introduction and Purpose

This document describes the Operations Support System Interface (OSSI) for the Modular Headend Architecture version 2 (MHAv2). MHAv2 is initially targeted to permit a CMTS to support an IP-based digital HFC plant. In an IP-based digital HFC plant, the fiber portion utilizes a baseband network transmission technology such as Ethernet, EPON (Ethernet Passive Optical Network), GPON (Gigabit Passive Optical Network), or any Layer 2 technology that would support a fiber-based layer 1.

MHAv2 uses a layer 3 pseudowire between a CCAP Core and a series of Remote PHY devices. One of the common locations for a Remote PHY device is at an optical node at the junction of the fiber and coax plants.

1.2 MHAv2 Interface Documents

A list of the documents in the MHAv2 family of specifications is provided below. For updates, refer to http://www.cablelabs.com/specs/specification-search/?cat=docsis&scat=dca-mhav2.

Designation	Title
[R-PHY]	Remote PHY Specification
[R-DEPI]	Remote Downstream External PHY Interface Specification
[R-UEPI]	Remote Upstream External PHY Interface Specification
[GCP]	Generic Control Plane Specification
[R-DTI]	Remote DOCSIS Timing Interface Specification
[R-OOB]	Remote Out-of-Band Specification
R-OSSI (this document)	Remote PHY Operations Support System Interface Specification

MHAv2 does not explicitly use the DTI specification or any of the MHA specifications.

1.3 Requirements

Throughout this document, the words that are used to define the significance of particular requirements are capitalized. These words are:

"MUST"	This word means that the item is an absolute requirement of this specification.
"MUST NOT"	This phrase means that the item is an absolute prohibition of this specification.
"SHOULD"	This word means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.
"SHOULD NOT"	This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
"MAY"	This word means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

This document defines many features and parameters, and a valid range for each parameter is usually specified. Equipment (CMTS and CCAP) requirements are always explicitly stated. Equipment complying with all mandatory

(MUST and MUST NOT) requirements is considered compliant with this specification. Support of non-mandatory features and parameter values is optional.

1.4 Conventions

In this specification the following convention applies any time a bit field is displayed in a figure. The bit field should be interpreted by reading the figure from left to right, then from top to bottom, with the MSB being the first bit so read and the LSB being the last bit so read.

```
MIB syntax, XML Schema and YANG module syntax are represented by this code sample font.
```

NOTE: Notices and/or Warnings are identified by this style font and label.

2 REFERENCES

2.1 Normative References

In order to claim compliance with this specification, it is necessary to conform to the following standards and other works as indicated, in addition to the other requirements of this specification. Notwithstanding, intellectual property rights may be required to use or implement such normative references.

[CANN]	CableLabs Assigned Names and Numbers, CL-SP-CANN-I13-150515, May 15, 2015, Cable Television Laboratories, Inc.
[CCAP-CONFIG-YANG]	CCAP YANG Configuration Module, ccap@2014-04-02.yang, http://www.cablelabs.com/YANG/DOCSIS
[CCAP-EVENTS-YANG]	CCAP YANG Module for Event Messaging, CCAPevents.yang, http://www.cablelabs.com/YANG/DOCSIS
[CCAP-OSSIv3.1]	DOCSIS 3.1 CCAP OSSI Specification, CM-SP-CCAP-OSSIv3.1-I04-150611, June 11, 2015, Cable Television Laboratories, Inc.
[CLAB-DEF-MIB]	CableLabs Definition MIB Specification, CL-SP-MIB-CLABDEF-I10-120809, August 9, 2012, Cable Television Laboratories, Inc.
[CLAB-TOPO-MIB]	CableLabs Topology MIB, CLAB-TOPO-MIB, http://www.cablelabs.com/MIBs/DOCSIS/.
[DOCS-DIAG-MIB]	DOCSIS Diagnostic Log MIB, DOCS-DIAG-MIB, http://www.cablelabs.com/MIBs/DOCSIS/.
[DOCS-IF3-MIB]	DOCSIS Interface 3 MIB Module, DOCS-IF3-MIB, http://www.cablelabs.com/MIBs/DOCSIS/.
[DOCS-IFEXT2-MIB]	DOCSIS Interface Extension 2 MIB Module, DOCS-IFEXT2-MIB, http://www.cablelabs.com/MIBs/DOCSIS/.
[DOCS-PNM-MIB]	DOCSIS PNM MIB Module, DOCS-PNM-MIB, http://www.cablelabs.com/MIBs/DOCSIS/.
[DOCS-RPHY-MIB]	DOCSIS Remote PHY MIB Module, DOCS-RPHY-MIB, http://www.cablelabs.com/MIBs/DOCSIS/.
[GCP]	Generic Control Plane Specification, CM-SP-GCP-I01-150615, June 15, 2015, Cable Television Laboratories, Inc
[L2VPN]	Layer 2 Virtual Private Networks, CM-SP-L2VPN-I15-150528, May 28, 2015, Cable Television Laboratories, Inc.
[MULPIv3.1]	MAC and Upper Layer Protocols Interface Specification, CM-SP-MULPIv3.1-I06-150611, June 11, 2015, Cable Television Laboratories, Inc.
[OSSIv3.0]	Operations Support System Interface Specification, CM-SP-OSSIv3.0-I26-150528, May 28, 2015, Cable Television Laboratories, Inc.
[PHYv3.1]	DOCSIS Physical Layer Specification, CM-SP-PHYv3.1-I06-150611, June 11, 2015, Cable Television Laboratories, Inc.
[R-DEPI]	Remote Downstream External PHY Interface Specification, CM-SP-R-DEPI-I01-150615, June 15, 2015, Cable Television Laboratories, Inc.
[R-DTI]	Remote DOCSIS Timing Interface Specification, CM-SP-R-DTI-I01-150615, June 15, 2015, Cable Television Laboratories, Inc.
[RFC 1350]	IETF RFC 1350/STD0033, The TFTP Protocol (Revision 2), July 1992.
[RFC 2560]	IETF RFC 2560, X.509 Internet Public Key Infrastructure Online Certification Status Protocol - OCSP, June 1999.
[RFC 2573]	IETF RFC 2573, SNMP Applications, April 1999.

[RFC 2575] IETF RFC 2575, View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP), April 1999.

[RFC 2578] IETF RFC 2578, Structure of Management Information Version 2 (SMIv2), April 1999.

[RFC 2669] IETF RFC 2669, DOCSIS Cable Device MIB Cable Device Management Information Base for DOCSIS compliant Cable Modems and Cable Modem Termination Systems, August 1999.

[RFC 2786] IETF RFC 2786, Diffie-Helman USM Key Management, March 2000.

[RFC 2790] IETF RFC 2790, Host Resources MIB, March 2000.

[RFC 2856] IETF RFC 2856, Textual Conventions for Additional High Capacity Data Types, June 2000.

[RFC 2863] IETF RFC 2863, The Interfaces Group MIB, June 2000.

[RFC 3164] IETF RFC 3164, The BSD Syslog Protocol, August 2001.

[RFC 3289] IETF RFC 3289, Management Information Base for the Differentiated Services Architecture, June 2002.

[RFC 3412] IETF RFC 3412, Message Processing and Dispatching for the Simple Network Management Protocol (SNMP), December 2002.

[RFC 3418] IETF RFC 3418/STD0062, Management Information Base (MIB) for the Simple Network Management Protocol (SNMP), December 2002.

[RFC 3433] IETF RFC 3433, Entity Sensor Management Information Base, December 2002.

[RFC 3584] IETF RFC 3584, Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework, August 2003.

[RFC 3635] IETF RFC 3635, Definitions of Managed Objects for the Ethernet-like Interface Types, October 2003.

[RFC 4022] IETF RFC 4022, Management Information Base for the Transmission Control Protocol (TCP), March 2005.

[RFC 4113] IETF RFC 4113, Management Information Base for the User Datagram Protocol (UDP), June 2005.

[RFC 4133] IETF RFC 4133, Entity MIB (Version 3), August 2005.

[RFC 4188] IETF RFC 4188, Definitions of Managed Objects for Bridges, September 2005.

[RFC 4250] IETF RFC 4250, The Secure Shell (SSH) Protocol Assigned Numbers, January 2006.

[RFC 4251] IETF RFC 4251, The Secure Shell (SSH) Protocol Architecture, January 2006.

[RFC 4252] IETF RFC 4252, The Secure Shell (SSH) Authentication Protocol, January 2006.

[RFC 4253] IETF RFC 4253, The Secure Shell (SSH) Transport Layer Protocol, January 2006.

[RFC 4254] IETF RFC 4254, The Secure Shell (SSH) Connection Protocol, January 2006.

[RFC 4293] IETF RFC 4293, Management Information Base for the Internet Protocol (IP), April 2006.

[RFC 4546] IETF RFC 4546, Radio Frequency (RF) Interface Management Information Base for Data over Cable Service Interface Specifications (DOCSIS) 2.0 Compliant RF Interfaces, June 2006.

[RFC 4639] IETF RFC 4639, Cable Device Management Information Base for Data-Over-Cable Service Interface Specification (DOCSIS) Compliant Cable Modems and Cable Modem Termination Systems, December 2006.

[RFC 5277] IETF RFC 5277, NETCONF Event Notifications, July 2008.

[RFC 5280] IETF RFC 5280, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, May 2008.

[RFC 5601] IETF RFC 5601, Pseudowire (PW) Management Information Base (MIB), July 2009.

[RFC 6021] IETF RFC 6021, Common YANG Data Types, October 2010.

[RFC 6991] IETF RFC 6991, Common YANG Data Types, July 2013.

[R-OOB] Remote Out-of-Band Specification, CM-SP-R-OOB-I01-150615, June 15, 2015, Cable Television Laboratories, Inc.

[R-PHY] Remote PHY System Specification, CM-SP-R-PHY-I01-150615, June 15, 2015, Cable Television Laboratories, Inc.

[R-UEPI] Remote Upstream External PHY Interface Specification, CM-SP-R-UEPI-I01-150615, June 15, 2015, Cable Television Laboratories, Inc.

[SCTE 154-2] ANSI SCTE 154-2 2008, SCTE-HMS-QAM-MIB.

[SCTE 154-5] ANSI SCTE 154-5 2008, SCTE-HMS-HEADENDIDENT TEXTUAL CONVENTIONS MIB.

[SECv3.1] DOCSIS 3.1 Security Specification, CM-SP-SECv3.1-I03-150611, June 11, 2015, Cable Television Laboratories, Inc.

[W3 XML1.0] Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation 04, February 2004.

[W3 XSD1.0] XML Schema Part 1: Structures Second Edition, W3C Recommendation 28, October 2004.

2.2 Informative References

This specification uses the following informative references.

[CCAP TR] Converged Cable Access Platform Architecture Technical Report, CM-TR-CCAP-V03-120511, May 11, 2012, Cable Television Laboratories, Inc.

[DRFI] DOCSIS Downstream RF Interface Specification, CM-SP-DRFI- I14-131120, November 20, 2013, Cable Television Laboratories, Inc.

[ISO 11404] BS ISO/IEC 11404:1996 Information technology--Programming languages, their environments and system software interfaces--Language-independent datatypes, January 2002.

[ISO 19501] ISO/IEC 19501:2005, Information technology - Open Distributed Processing - Unified Modeling Language (UML) Version 1.4.2.

[ITU-T X.692] ITU-T Recommendation X.692 (03/2002), Information technology - ASN.1 encoding rules: Specification of Encoding Control Notation (ECN).

[ITU-T M.3400] ITU-T Recommendation M.3400 (02/2000): TMN AND Network Maintenance: International Transmission Systems, Telephone Circuits, Telegraphy, Facsimile and Leased Circuits, TMN management functions.

[M-OSSI] DOCSIS M-CMTS Operations Support Interface, CM-SP-M-OSSI-I08-081209, December 9, 2008, Cable Television Laboratories, Inc.

[NSI] Cable Modem Termination System - Network Side Interface Specification, SP-CMTS-NSI-I01-960702, July 2, 1996, Cable Television Laboratories, Inc.

[PMI] Edge QAM Provisioning and Management Interface Specification, CM-SP-EQAM-PMI-I02-111117, November 17, 2011, Cable Television Laboratories, Inc.

[RFC 791] IETF RFC 791, Internet Protocol, September 1981.

[RFC 1042] IETF RFC 1042/STD0043, Standard for the transmission of IP datagrams over IEEE 802 networks, February 1988.

[RFC 1123] IETF RFC 1123/STD0003, Requirements for Internet Hosts - Application and Support, October 1989.

[RFC 1157] IETF RFC 1157, Simple Network Management Protocol (SNMP), May 1990.

[RFC 1213] IETF RFC 1213/STD17, Management Information Base for Network Management of TCP/IP-based internets: MIB-II, March 1991.

[RFC 1901] IETF RFC 1901, Introduction to Community-based SNMPv2, January 1996.

[RFC 2579] IETF RFC 2579, Textual Conventions for SMIv2, April 1999.

[RFC 2580] IETF RFC 2580, Conformance Statements for SMIv2, April 1999.

[RFC 3260] IETF RFC 3260, New Terminology and Clarifications for Diffserv, April 2002.

[RFC 3339] IETF RFC 3339, Date and Time on the Internet: Timestamps, July 2002.

[RFC 3410] IETF RFC 3410, Introduction and Applicability Statements for Internet-Standard Management Framework, December 2002.

[RFC 3411] IETF RFC 3411/STD0062, An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks, December 2002.

[RFC 3413] IETF RFC 3413, Simple Network Management Protocol (SNMP) Applications, December 2002.

[RFC 3414] IETF RFC 3414/STD0062, User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3), December 2002.

[RFC 3415] IETF RFC 3415, View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP), December 2002.

[RFC 3416] IETF RFC 3416, Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP), December 2002.

[RFC 3417] IETF RFC 3417, Transport Mappings for the Simple Network Management Protocol (SNMP), December 2002.

[RFC 3419] IETF RFC 3419, Textual Conventions for Transport Addresses, December 2002.

[RFC 4001] IETF RFC 4001, Textual Conventions for Internet Network Addresses, February 2005.

[RFC 4181] IETF RFC 4181, Guidelines for Authors and Reviewers of MIB Documents, September 2005.

[RFC 4291] IETF RFC 4291, IP Version 6 Addressing Architecture, February 2006.

[RFC 6020] IETF RFC 6020, YANG - A data modeling language for the Network Configuration Protocol (NETCONF), October 2010.

2.3 Reference Acquisition

- Cable Television Laboratories, Inc., 858 Coal Creek Circle, Louisville, CO 80027; Phone +1-303-661-9100; Fax +1-303-661-9199; http://www.cablelabs.com
- IANA, Internet Assigned Numbers Authority (IANA); http://www.iana.org
- IETF, Internet Engineering Task Force (IETF) Secretariat, 48377 Fremont Blvd., Suite 117, Fremont, California 94538, USA; Phone: +1-510-492-4080, Fax: +1-510-492-4001; http://www.ietf.org/
- ISO Specifications, International Organization for Standardization (ISO), 1, rue de Varembé, Case postale 56, CH-1211 Geneva 20, Switzerland; Phone +41 22 749 01 11; Fax +41 22 733 34 30; http://www.iso.org
- ITU Recommendations, International Telecommunication Union, Place des Nations, CH-1211, Geneva 20, Switzerland; Phone +41-22-730-51-11; Fax +41-22-733-7256; http://www.itu.int
- SCTE, Society of Cable Telecommunications Engineers Inc., 140 Philips Road, Exton, PA 19341; Phone: 1+610-363-6888 /1+ 800-542-5040; Fax: 1+610-363-5898; http://www.scte.org/
- World Wide Web Consortium (W3C), Massachusetts Institute of Technology, 32 Vassar Street, Room 32-G515, Cambridge, MA 02139; Phone +1-617-253-2613, Fax +1-617-258-5999; http://www.w3.org/Consortium/

3 TERMS AND DEFINITIONS

This specification uses the following terms:

Aggregation	A special type of object association for Configuration Object Models in which objects are assembled or configured together to create a more complex object.
Bonded Channels	A logical channel comprising multiple individual channels.
Bridging CMTS	A CMTS that makes traffic forwarding decisions between its Network Systems Interfaces and MAC Domain Interfaces based upon the Layer 2 Ethernet MAC address of a data frame.
Cable Modem	A modulator-demodulator at subscriber locations intended for use in conveying data communications on a cable television system.
Cable Modem Termination System	An access-side networking element or set of elements that includes one or more MAC Domains and one or more Network System Interfaces. This unit is located at the cable television system headend or distribution hub and provides data connectivity between a DOCSIS Radio Frequency Interface and a wide-area network.
Cable Modem Termination System - Network Side Interface (CMTS-NSI)	The interface, defined in [NSI], between a CMTS and the equipment on its network side.
Carrier-to-Noise plus Interference Ratio (CNIR)	The ratio of the expected commanded received signal power at the CMTS input to the noise plus interference in the channel.
CCAP Core	A CCAP device which uses MHAv2 protocols to interconnect to R-PHY Entity devices.
Channel	The frequency spectrum occupied by a signal. Usually specified by center frequency and bandwidth parameters.
Command Line Interface	A mechanism used to interact with the CCAP by typing text-based commands into a system interface.
Configuration Objects	Managed objects in the CCAP configuration that support writeability. The CCAP is configured by specifying the attributes of these objects.
Converged Cable Access Platform	An access-side networking element or set of elements that combines the functionality of a CMTS with that of an Edge QAM, providing high-density services to cable subscribers.
Converged Interconnect Network	The network (generally gigabit Ethernet) that connects a CCAP Core to an R-PHY Entity.
Customer Premises Equipment	Equipment at the end user's premises; may be provided by the service provider.
Downstream	1. Transmissions from CMTS to CM. This includes transmission from the CCAP Core to the RPD, as well as the RF transmissions from the RPD to the CM. 2. RF spectrum used to transmit signals from a cable operator's headend or hub site to subscriber locations.
Extensible Markup Language	A universal file format for storing and exchanging structured data. The CCAP configuration file is created in XML and has a specific schema, generated from a set of YANG modules, which are a physical implementation of an object model created to describe CCAP configuration.
FCAPS	A set of principles for managing networks and systems, wherein each letter represents one principle. F is for Fault, C is for Configuration, A is for Accounting, P is for Performance, and S is for Security.

Flow	A stream of packets in DEPI used to transport data of a certain priority from the CCAP Core to a particular QAM channel of the R-PHY Entity. In PSP operation, there can exist several flows per QAM channel.
Generalization	A relationship in which one configuration model element (the child) is based on another model element (the parent). A generalization relationship indicates that the child receives all of the attributes, operations, and relationships that are defined in the parent.
Hybrid Fiber/Coax System	A broadband bidirectional shared-media transmission system using optical fiber trunks between the headend and the fiber nodes, and coaxial cable distribution from the fiber nodes to the customer locations.
Institute of Electrical and Electronic Engineers	A voluntary organization which, among other things, sponsors standards committees and is accredited by the American National Standards Institute (ANSI).
Internet Engineering Task Force	A body responsible for, among other things, developing standards used in the Internet.
Internet Protocol	An Internet network-layer protocol.
L2TP Pseudowire (PW)	An emulated circuit as it traverses a packet-switched network. There is one Pseudowire per L2TP Session.
L2TP Pseudowire Type	The payload type being carried within an L2TP session. Examples include PPP, Ethernet, and Frame Relay.
L2TP Session	An L2TP session is the entity that is created between two LCCEs in order to exchange parameters for and maintain an emulated L2 connection. Multiple sessions may be associated with a single Control Connection.
MAC Domain	A grouping of Layer 2 devices that can communicate with each other without using bridging or routing. In DOCSIS, it is the group of CMs that are using upstream and downstream channels linked together through a MAC forwarding entity.
MAC Domain Cable Modem Service Group	The subset of a Cable Modem Service Group which is confined to the Downstream Channels and Upstream Channels of a single MAC domain. Differs from a CM-SG only if multiple MAC domains are assigned to the same CM-SGs.
Management	Functions on the CCAP that monitor for faults and for overall system performance, including traps and alarms.
Media Access Control	Used to refer to the Layer 2 element of the system which would include DOCSIS framing and signaling.
Management Information Base	A database of device configuration and performance information which is acted upon by SNMP.
Multiple System Operator	A corporate entity that owns and/or operates more than one cable system.
Open Systems Interconnection (OSI)	A framework of ISO standards for communication between different systems made by different vendors, in which the communications process is organized into seven different categories that are placed in a layered sequence based on their relationship to the user. Each layer uses the layer immediately below it and provides a service to the layer above. Layers 7 through 4 deal with end-to-end communication between the message source and destination, and layers 3 through 1 deal with network functions.
Physical (PHY) Layer	Layer 1 in the Open System Interconnection (OSI) architecture; the layer that provides services to transmit bits or groups of bits over a transmission link between open systems and which entails electrical, mechanical and handshaking procedures.

Quadrature Amplitude Modulation	A modulation technique in which an analog signal's amplitude and phase vary to convey information, such as digital data.
QAM Channel	Analog RF channel that uses quadrature amplitude modulation (QAM) to convey information.
Radio Frequency	In cable television systems, this refers to electromagnetic signals in the range 5 to 1000 MHz.
Remote PHY Device	The Remote PHY Device contains mainly PHY related circuitry, such as downstream QAM modulators, upstream QAM demodulators, and pseudowire logic to connect to the CCAP Core. Together, the CCAP Core and the R-PHY Entity are the functional equivalent of an I-CMTS (Integrated CMTS), just with different packaging.
Request for Comments	A technical policy document of the IETF; these documents can be accessed at http://www.rfc-editor.org/.
Routing CMTS	A CMTS that makes traffic forwarding decisions between its Network System Interfaces and MAC Domain Interfaces based upon the Layer 3 (network) address of a packet.
Running-config	Configuration objects that control CCAP behavior, along with any vendor-proprietary configurations.
Secure Copy Protocol	A secure file transfer protocol based on Secure Shell (SSH).
Simple Network Management Protocol	Allows a host to query modules for network-related statistics and error conditions.
Specialization	A relationship in which one configuration model element (the parent) is used to model another element (the child). The specialized child element receives all of the attributes, operations, and relationships that are defined in the parent and defines additional attributes, operations and relationships that enable its specialized behavior.
Startup-config	The configuration objects stored in non-volatile memory.
Upstream	1. Transmissions from CM to CCAP. This includes transmission from the RPD to the CCAP Core, as well as the RF transmissions from the CM to the RPD. 2. RF spectrum used to transmit signals from a subscriber location to a cable operator's headend or hub site.
X.509	ITU-T Recommendation standard for a public key infrastructure (PKI) for single sign-on (SSO) and Privilege Management Infrastructure (PMI).
YANG	A data modeling language for the NETCONF network configuration protocol. Though the CCAP physical data model for configuration makes use of one or more YANG modules, NETCONF implementation is not required for the integrated CCAP.

4 ABBREVIATIONS, ACRONYMS, AND NAMESPACES

This specification uses the following abbreviations:

AAA	Network Authentication, Authorization, and Accounting
ACL	Access Control List
AQM	Active Queue Management
AVP	Attribute Value Pair
BPI	Baseline Privacy Interface
BSS	Business Support Systems
CA	Certificate Authority
CCAP	Converged Cable Access Platform
CIN	Converged Interconnect Network
CLI	Command Line Interface
CM	Cable Modem
CMTS	Cable Modem Termination System
CPAF	Configuration, Performance, Accounting, Fault Management
CPE	Customer Premises Equipment
CRL	Certificate Revocation List
CW	Control Word
DBG	Downstream Bonding Group
DCID	Downstream Channel Identifier
DCS	Downstream Channel Sets
DEPI	Downstream External PHY Interface
DHCP	Dynamic Host Configuration Protocol
DLC	Downstream Line Card
DPoE	DOCSIS Provisioning of EPON
DS	Downstream
DSID	Downstream Service ID
EAE	Early Authentication and Encryption
EPON	Ethernet Passive Optical Network
EQAM	Edge QAM
ERM	Edge Resource Manager
ERMI	Edge Resource Manager Interface
ERRP	Edge Resource Registration Protocol
FCAPS	Fault, Configuration, Accounting, Performance and Security
FQDN	Fully Qualified Domain Name
FRU	Field Replaceable Unit
GCP	Generic Control Plane
HFC	Hybrid Fiber/Coax System
HTTPS	Secure Hypertext Transfer Protocol
I-CMTS	Integrated CMTS

IEEE	Institute of Electrical and Electronic Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
ITU	International Telecommunication Union
L2VPN	Layer 2 Virtual Private Network
MAC	Media Access Control
MD-CM-SG	Media Access Control Domain Cable Modem Service Group
MDD	MAC Domain Descriptor
MIB	Management Information Base
MPEG	Moving Picture Experts Group
MPEG-TS	Moving Picture Experts Group-Transport Stream
MPT	MPEG-TS mode of DEPI
MPTS	Multi-Program Transport Stream
MSO	Multiple System Operator
MTC	Multiple Transmit Channel
MTU	Maximum Transmission Unit
NMS	Network Management System
NOC	Network Operations Center
NSI	Network Side Interface
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiplexing with Multiple Access
OM	Object Model (Information Model)
OMG	Object Management Group
OOA&D	Object-Oriented Analysis and Design
OSS	Operations Support System
OSSI	Operations Support System Interface
OUI	Organization Unique Identifier
PAT	Program Association Table
PEN	Private Enterprise Number
PID	Packet Identifier
PLC	Phy Link Channel
PW	Pseudowire
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RCC	Receive Channel Configuration
RCP	Receive Channel Profile
RDTI	Remote DTI
RF	Radio Frequency
RFC	Request for Comments
RPD	Remote PHY Device

R-PHY	Remote PHY
SA	Security Association
SCP	Secure Copy Protocol
SDV	Switched Digital Video
SMIv2	Structure of Management Information Version 2
SNMP	Simple Network Management Protocol
SNMPv1	Version 1 of the Simple Network Management Protocol
SNMPv2	Version 2 of the Simple Network Management Protocol
SNMPv3	Version 3 of the Simple Network Management Protocol
SSH	Secure Shell
SSM	Source Specific Multicast
STP	Spanning Tree Protocol
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
TLS	Transport Layer Security
TLV	Type Length Value Attribute
ToD	Time of Day
ToS	Terms of Service
TS	Transport Stream
TSID	Transport Stream Identifier
UBG	Upstream Bonding Group
UCID	Upstream Channel Identifier
UDC	Upstream Drop Classifier
UDP	User Datagram Protocol
ULC	Upstream Line Card
UML	Unified Modeling Language
URL	Uniform Resource Locator
US	Upstream
VLAN	Virtual Local Area Network
XML	Extensible Markup Language
XSD	XML Schema Definition

5 OVERVIEW

5.1 FCAPS Network Management Model

The International Telecommunication Union (ITU) Recommendation [ITU-T M.3400] defines a set of management categories, referred to as the FCAPS model, represented by the individual management categories of Fault, Configuration, Accounting, Performance and Security. Telecommunications operators, including MSOs, commonly use this model to manage large networks of devices. This specification uses these management categories to organize the requirements for the configuration and management of the Remote PHY platform.

Fault management seeks to identify, isolate, correct and record system faults. Configuration management modifies system configuration variables and collects configuration information. Accounting management collects usage statistics for subscribers, sets usage quotas and bills users according to their use of the system. Performance management focuses on the collection of performance metrics, analysis of these metrics and the setting of thresholds and rate limits. Security management encompasses identification and authorization of users and equipment, provides audit logs and alerting functions, as well as providing vulnerability assessment.

Each of these management categories is discussed in further detail in [CCAP-OSSIv3.1].

5.2 Management Architectural Overview

Figure 5-1 illustrates the Remote PHY management architecture. The CM, RPD and CCAP Core reside within the Network Layer where services are provided to end Subscribers and various metrics are collected about network and service performance, among other things. Various management servers reside in the Network Management Layer within the MSO back office to provision, monitor and administer the Network Elements within the Network Layer.

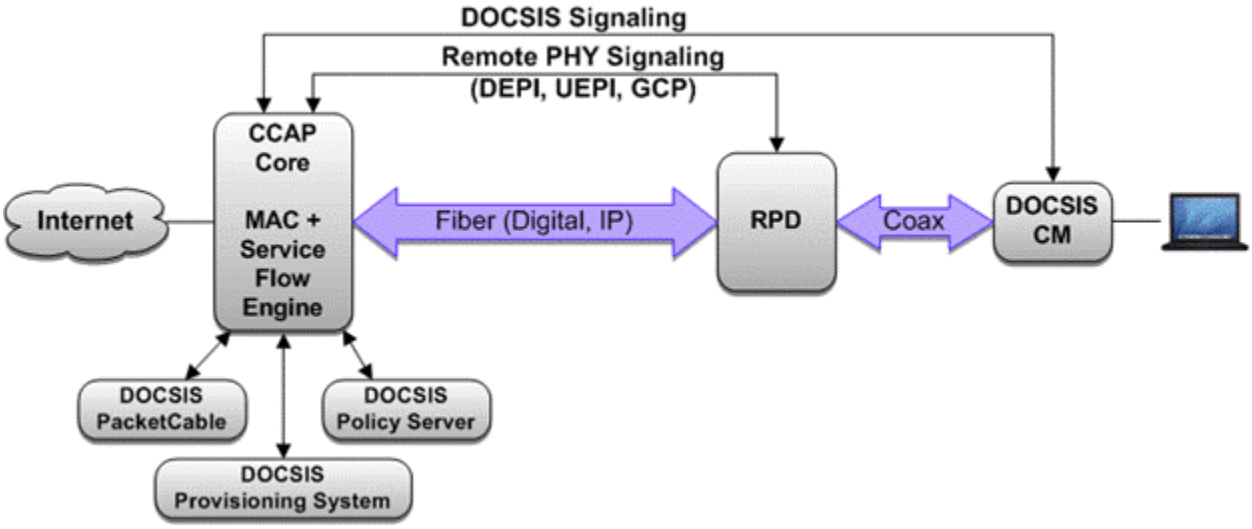

Figure 5-1 - CCAP Configuration Objects

Finally, the Business and Service Management Layer is where higher level MSO business processes are implemented via BSS/OSS systems. These BSS/OSS systems utilize the data and information from the Network Management Layer which interrogate data from the Network Layer.

5.3 Remote PHY OSSI Key Features

The primary goals of the Remote PHY OSSI are:

- Ensure that existing management interfaces on the CCAP are supported transparently to the NMS.

- Ensure that the RPD can be configured and managed both indirectly via the CCAP Core and, when necessary, directly.

Table 5-1 - Management Feature Requirements for Remote PHY

Features	Management Functional Area	OSI Layer	Description
RPD Configuration	Configuration	PHY, Data Link	Provisioning physical downstream and upstream interfaces and other features on the RPD indirectly via the CCAP Core.
CCAP Core Remote Phy Configuration	Configuration	PHY, Data Link	Configuration of the Remote PHY feature set and the RPD on the CCAP Core.
RPD Fault Detection	Fault	PHY, Network	Remote PHY Device fault definition and detection
CCAP Core Remote PHY Fault Detection	Fault	PHY, Network	Detection of faults related to the Remote PHY feature set on the CCAP Core.
RPD Performance/Status	Performance	PHY, Network	Interface for a management agent on the RPD to communicate non-DOCSIS performance and status information.
CCAP Core Performance/Status	Performance	PHY, Network	Monitoring of the DOCSIS Phy (indirectly via the RPD) and direct monitoring of non-DOCSIS performance and status information related to the Remote PHY feature set.

5.3.1 Fault Management Features

The Remote PHY Fault Management requirements include:

- Extended lists of events related to the new set of Remote PHY features for both the CCAP Core and RPD.
- Management requirements for the RPD to be managed directly, including an SNMP management agent (and associated MIBs) required on the RPD itself.
- Remote PHY Device (RPD) fault management requirements.
- Ensuring that existing faults defined for the CCAP function transparently in a Remote PHY environment.

5.3.2 Configuration Management Features

The configuration of the RPD by the CCAP Core is defined in this specification. The reporting of configuration state and status information is done via SNMP MIB objects. Configuration of features and functions of the CCAP is performed via XML configuration files.

The Remote PHY configuration requirements include:

- Configuration management of the CCAP Core as it relates to the Remote PHY Device and Remote PHY features.
- Configuration management of the Remote PHY by the CCAP Core.

5.3.3 Performance Management Features

The Remote PHY performance management requirements include:

- Non-DOCSIS performance management objects defined on the RPD and CCAP Core to monitor the performance and status of the Remote PHY feature.
- Ensuring that existing performance management objects defined for the CCAP function transparently in a Remote PHY environment.

5.4 Information Models

The Information Model approach is based on an object-oriented modeling approach well known in the industry for capturing requirements and analyzing the data in a protocol independent representation. This approach defines

requirements with use cases to describe the interactions between the operations support systems and the network element. The management information is represented in terms of objects along with their attributes and the interactions between these encapsulated objects (or also referred to as entities in some representations). The diagrams developed to capture these managed objects and their attributes and associations are UML Class Diagrams. The collection of UML Class Diagrams and Use Case Diagrams are referred to as the Remote PHY Information Models. With the introduction of several new, complex features in Remote PHY and the operator needs for a more proactive and efficient approach to management information, information modeling methodologies offer the ability to reuse the same definitions when new protocols are introduced in the future.

The managed objects are then represented in a protocol-specific form referred to as a Management Data Model. The Management Data Models when using SNMP are described using the Structure of Management Information Version 2 (SMIv2) [RFC 2578] and the design of these models is determined by the capabilities of the protocol. The Management Data Models when using XML configuration file download are described using XML Schema [W3 XSD1.0]. The Management Data Models when using GCP are defined as TLVs.

5.5 CCAP-OSSI Document Organization

This specification uses the FCAPS framework to group topics and content. In order to provide a more logical flow, one that mirrors processes in place at MSOs, the order of functions has been shifted and is organized as CPAF:

- Configuration Management
- Performance Management
- Accounting Management
- Fault Management

Note that Security Management topics are covered in context of these topics.

6 INFORMATION MODELING FOR OSSI

6.1 Information Model Notation

The Unified Modeling Language (UML) is a unified model for object-oriented analysis and design (OOA&D). UML is an OMG standard and is an accepted ISO specification [ISO 19501].

UML defines a general-purpose, graphical modeling language that can be applied to any application domain (e.g., communications) and implementation platforms (e.g., J2EE).

The OSSI Information Model diagram is represented by the UML Class Diagram. The class diagram describes the types of objects existing in a system and their static relationship or association.

6.1.1 Classes

Classes are generally represented by a square box with three compartments. The top compartment contains the class name (used here as the object name) with the first letter capitalized. The middle compartment contains the list of attributes with the first letter of each attribute in lower case. The bottom compartment contains the list of operations. For the purposes of this specification, the methods section of the class box is not used (suppressed) and the implementation level details of the attributes are omitted.

Attributes also include a visibility notation which precedes the attribute name and is one of the following:

- '+' public (default)
- '-' private
- '#' protected

If the above notation is omitted from the attribute, the default of public is implied. For the purposes of this specification, the protected visibility generally refers to indexes of MIB tables, schema instances, etc.

An interface is represented in the class diagram as an object with the keyword <<interface>> preceding the object name. In general, an interface is a declaration of a set of public features and obligations (such as get methods).

6.1.2 Associations

A class diagram also contains associations which represent relationships between instances of classes. An association has two ends with each end attached to one of the classes. The association end also has a multiplicity indicator which defines how many objects may participate in the relationship. Multiplicity notation is as follows:

- '1' exactly one
- '*' zero or more (default)
- '0..1' zero or one (optional)
- 'm..n' numerically specified

If the above notation is omitted from the association end, the default of '*' is implied.

If one end of the association contains an open arrowhead, this implies navigability in the direction indicated by the arrow.

6.1.3 Generalization

Generalization is the concept of creating subclasses from superclasses and is also known as inheritance within programming languages. Subclasses include (or inherit) all the elements of the superclass and may override inherited methods. Subclasses are more specific classes while superclasses are generalized classes.

The UML notation for Generalization is shown as a line with a hollow triangle as an arrowhead pointing to the generalized class.

6.1.4 Dependencies

Dependencies between two classes are represented by a dashed arrow between two objects. The object at the tail of the arrow depends on the object at the other end.

6.1.5 Comment

A Comment in a class diagram is a textual annotation attached to any element. This is represented as a note symbol with a dashed line connecting the note with the element.

6.1.6 Diagram Notation

Figure 6-1 highlights the UML Class Diagram notation discussed in this section. Figure 6-1 is not a complete representation of the UML Class Diagram notation, but captures those concepts used throughout this specification.

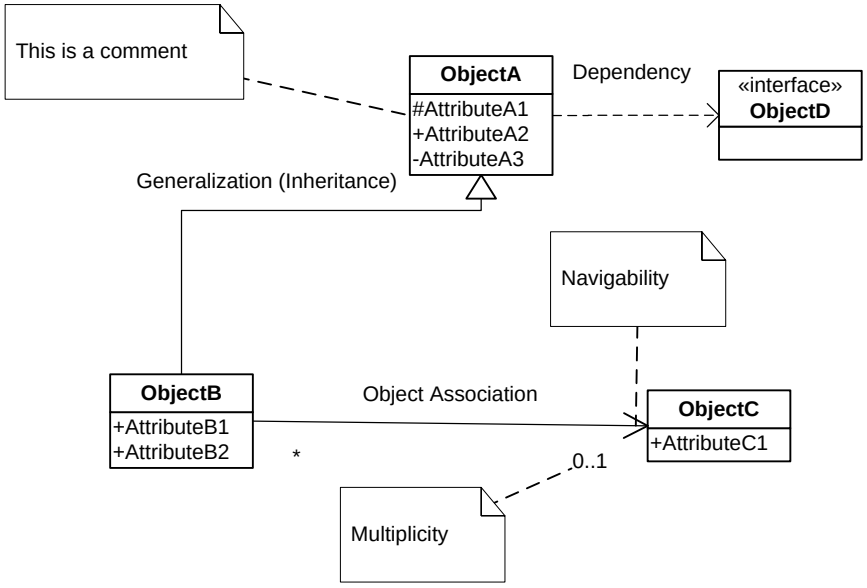

Figure 6-1 - Object Model UML Class Diagram Notation

6.2 Object Instance Diagram

An Object Instance Diagram represents the objects in a system during one snapshot in time. In this diagram, the class objects are instantiated.

Figure 6-2 shows an Object Instance Diagram for an instantiation (myObjectA) of ObjectA from Figure 6-1.

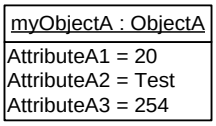

Figure 6-2 - Object Instance Diagram for ObjectA

6.3 ObjectA Definition Example

This section defines the details of the object and its associated attributes as defined in the object model diagram. The description of the object includes behavior, persistence requirements (if any), object creation and deletion behavior (if any), etc.

Table 6-1 lists the attributes the object defines in the object model. The object table is derived from the object model diagram where each row in the table represents an attribute of the object.

The "Attribute Name" column contains each defined attribute of the object. The naming convention for attributes is to capitalize the first letter and each letter of successive words within the name. Also, attribute names typically do not include any of the object name elements since this would cause duplication when the object and attributes are realized in SNMP.

The "Type" column contains the data type for the attribute. The data type can be a simple type such as unsignedInt or a defined data type such as EnumBits. DOCSIS 3.0 data types are defined in Section 6.5.

The "Access" column indicates the attributes accessibility (as mapped to an SNMP object for example). Example values include "key", "read-only", "read-write", and "read-create".

The "Type Constraints" column lists constraints on the normal data type specified in the "Type" column. If there are no defined constraints for the attribute, this column is empty. The example below for AttributeA1 lists a constraint on the unsignedInt Type where the range starts from 1 instead of normally starting from 0 for an unsignedInt.

The "Units" column lists units for the attribute or "N/A" if the attribute does not have units.

The "Default" column contains the default value for the attribute or "N/A" if the attribute does not have a default value or in cases where the attribute's description defines rules for the initialization value.

The sections following Table 6-1 are attribute descriptions which might include behavioral requirements or references.

Table 6-1 - ObjectA Example Table Layout

Attribute Name	Type	Access	Type Constraints	Units	Default
AttributeA1	unsignedInt	key	1..4294967295	N/A	N/A
AttributeA2	AdminString	read-write	SIZE (1..15)	N/A	N/A
AttributeA3	unsignedByte	read-create		seconds	60

6.3.1 AttributeA1

AttributeA1 is a key defined for…

NOTE: Objects which represent a table (in an SNMP MIB realization) and have N number of instances need to include at least one "key" attribute which is used to denote the instance or id. Key attributes are typically denoted with a protected visibility whereas all other attributes are denoted with a public visibility.

6.3.2 AttributeA2

AttributeA2 is …

NOTE: Persistence requirements are documented at the object level, not at the attribute level.

6.3.3 AttributeA3

AttributeA3 is …

6.4 Common Terms Shortened

The following table lists common terms which have been shortened to allow shorter SNMP MIB names. These shortened names are desired to be used consistently throughout the object models, SNMP MIBs and IPDR schemas. However, in some cases it might not be possible to maintain parity with pre-3.0 DOCSIS requirements.

Table 6-2 - Shortened Common Terms

Original Word	Shortened Word
Address	Addr
Aggregate	Agg
Algorithm	Alg
Application	App
Attribute	Attr
Authorization	Auth
Channel	Ch
Command	Cmd
Config*	Cfg
Control	Ctrl
Default	Def
Destination	Dest
Direction	Dir
Downstream	Ds
Encryption	Encrypt
Equalization	Eq
Group	Grp
Length	Len
Maximum	Max
Minimum	Min
Multicast	Mcast
Provision*	Prov
Receive	Rx
Registration	Reg
Replication	Repl
Request	Req
Resequence	Reseq
Resequencing	Reseq
Response	Rsp
Segment	Sgmt
Sequence	Seq
Service	Svc
ServiceFlow	Sf
Session(s)	Sess
Source	Src
Threshold	Thrshld
Total	Tot
Transmit	Tx
Upstream	Us
* indicates a wildcard	

6.4.1 Exceptions

Data types and managed objects do not consistently use the shortened names. Also, the term ServiceFlowId remains unchanged. Service and ServiceFlow are often not shortened to retain backward compatibility with QoS managed objects.

6.5 Data Types

This section includes the data type definitions for the Information Models defined for use in the CCAP. UML is used for modeling the management requirements.

The data types defined in this section are mapped for use with SNMP MIBs, IPDR XML schemas, YANG modules and XSD Schemas.

6.5.1 Data Types Mapping

XML is becoming the standard for data definition models. With XML, data transformations can be done with or without a model (DTD or Schema definition). DTDs and XML schemas provide additional data validation layer to the applications exchanging XML data. There are several models to map formal notation constructs like ASN.1 to XML [ITU-T X.692], UML to XML, YANG to XML, or XML by itself can be used for modeling purposes.

Each area of data information interest approaches XML and defines data models and/or data containment structures and data types. Similarly, SNMP took and modified a subset of ASN.1 for defining the Structured Management Information SMIv1 and SMIv2.

Due to the lack of a unified data model and data types for Network Management, a neutral model would be appropriate to allow capturing specific requirements and methodologies from existing protocols and allow forward or reverse engineering of those standards like SNMP to the general object model and vice versa.

6.5.2 Data Types Requirements and Classification

The Information Model has to provide seamless translation for SMIv2 requirements, in particular when creating MIB modules based on the Information Model. This specification needs to provide full support of [RFC 2578], [RFC 2579], and the clarifications and recommendations of [RFC 4181].

The Information Model has to provide seamless translation for YANG modeling requirements, in particular when creating YANG modules based on the Information Model.

Thus, there are two data type groups defined for modeling purposes and mapping to protocol data notation roundtrip.

- General data types

  Required data types to cover all the management syntax and semantic requirement for all OSSI supported data models. In this category are data types defined in SNMP SMIv2 [RFC 2578], and YANG common data types [RFC 6991].

- Extended data types

  Management protocols specialization based on frequent usage or special semantics. Required data types to cover all the syntax requirement for all OSSI supported data models. In this category are SNMP TEXTUAL-CONVENTION clauses [RFC 2579] of mandatory or recommended usage by [RFC 2579] and [RFC 4181] when modeling for SNMP MIB modules.

6.5.3 Data Type Mapping Methodology

The specification "XML Schema Part 2: Data types Second Edition" is based on [ISO 11404], which provides language-independent data types (see XML Schema reference). The mapping proposed below uses a subset of the XML schema data types to cover both SNMP forward and reverse engineering, and IPDR types. Any additional protocol being added should be feasible to provide the particular mappings.

SMIv2 has an extensive experience of data types for management purposes; for illustration consider Counter32 and Counter64 SMIv2 types [RFC 2578]. The XML schema data types makes no distinction of derived 'decimal' types and the semantics that are associated to counters, e.g., counters do not necessarily start at 0.

Most of the SNMP information associated to data types are reduced to size and range constraints and specialized enumerations.

6.5.4 General Data Types (SNMP Mapping)

Table 6-3 represents the mapping between the OSSI object model General Types and their equivalent representation for SNMP MIB Modules and IPDR Service Definitions. The permitted values for the data types are indicated in terms of value ranges and string length when applicable. The OM Data Type column includes the data types to map to SNMP, using the appropriated type in the corresponding protocol if applicable or available. The SNMP Mapping references to SNMP data types are defined in [RFC 2578] or as described below.

Note that SNMP does not provide float, double or long XML-Schema data types. Also, SNMP might map a type to an SNMP subtyped value. For example, unsignedByte data type maps to Unsigned32 subtyped to the appropriate range indicated by the Permitted Values (0..255 in this case). Other data types are mapped to SNMP TEXTUAL-CONVENTIONS as indicated by the references.

Table 6-3 - General Data Types

OM Data Type	XML-Schema data type	Permitted Values	SNMP Mapping
Enum	int	-2147483648..2147483647	INTEGER
EnumBits	hexBinary		BITS
Int	int	-2147483648..2147483647	Integer32
unsignedInt	unsignedInt	0..4294967295	Unsigned32
long	long	-9223372036854775808..-9223372036854775807	N/A
unsignedLong	unsignedLong	0..18446744073709551615	CounterBasedGauge64 [RFC 2856]
hexBinary	hexBinary		OCTET STRING
string	string		SnmpAdminString [RFC 3411]
boolean	boolean		TruthValue [RFC 2579]
Byte	byte	-128..127	Integer32
unsignedByte	unsignedByte	0..255	Unsigned32
Short	short	-32768..32767	Integer32
unsignedShort	unsignedShort	0..65535	Unsigned32
Gauge32	unsignedInt		Gauge32
Counter32	unsignedInt		Counter32
Counter64	unsignedLong		Counter64
IpAddress	hexBinary	SIZE (4)	IpAddress
Opaque	hexBinary		Opaque
dateTime	dateTime		DateAndTime
dateTimeMsec	unsignedLong		CounterBasedGauge64 [RFC 2856]
InetAddressIPv4	hexBinary	SIZE (4)	InetAddressIPv4 [RFC 4001]
InetAddressIPv6	hexBinary	SIZE (16)	InetAddressIPv6 [RFC 4001]
InetAddress			InetAddress [RFC 4001]
InetAddressType			InetAddressType [RFC 4001]
Uuid	hexBinary		OCTET STRING
MacAddress	hexBinary	SIZE (6)	MacAddress

6.5.5 Primitive Data Types (YANG Mapping)

Table 6-4 represents the mapping between the CCAP primitive data types and their equivalent representation in YANG. The permitted values for the data types are indicated in terms of value ranges and string length when applicable. The UML Primitive Data Type column includes the data types to map to YANG, using the appropriate type in YANG. The YANG Built-In Data Type Mapping references YANG data types defined in [RFC 6021] or as described below.

Table 6-4 - Primitive Data Types

UML Primitive Data Type	YANG Data Type Mapping	Permitted Values
HexBinary	ccap-octet-data-type	([0-9a-fA-F]{2})*
EnumBits	bits	
Boolean	boolean	true, false
Enum	enumeration	-2147483648..2147483647
Byte	int8	-128..127
Short	int16	-32768..32767
Integer	int32	-2147483648..2147483647
Long	int64	-9223372036854775808..9223372036854775807
String	string	
UnsignedByte	uint8	0..255
UnsignedShort	uint16	0..65535
UnsignedInt	uint32	0..4294967295
UnsignedLong	uint64	0..18446744073709551615

6.5.6 Extended Data Types (SNMP Mapping)

There are two sources of Extended Data Types: Protocol specific data types, and OSSI data types.

SNMP derived types are defined in SNMP MIB Modules. The most important are in [RFC 2579], which is part of SNMP STD 58, and are considered in many aspects part of the SNMP protocol. Other MIB modules TEXTUAL-CONVENTION definitions have been adopted and recommended (e.g., [RFC 4181]) for re-usability and semantics considerations in order to unify management concepts; some relevant RFCs that include commonly used textual conventions are [RFC 4001], [RFC 2863], [RFC 3411], and [RFC 3419] among others (see [RFC 4181]).

Table 6-5 includes the most relevant data types taken from SNMP to provide a direct mapping of the OSSI object model to SNMP MIB modules. For example, TagList comes from [RFC 3413] SnmpTaglist and preserves its semantics; AdminString comes from [RFC 3411] SnmpAdminString.

In general, when an OSSI object model needs to reference an existing SNMP textual convention for the purpose of round trip design from UML to SNMP, these textual conventions can be added to this list. Other sources of textual conventions not listed here are from MIB modules specific to DOCSIS, either as RFCs or Annex documents in this specification. Some of those sources are [RFC 4546] and Annex A.

OSSI data types are also defined in this specification in the Data Type section of OSSI annexes; for example, Annex A.

Table 6-5 - Extended Data Types

OM Data Type	XML-Schema data type	Permitted Values	SNMP Mapping
PhysicalIndexOrZero	unsignedInt	0..2147483647	Integer32
TagList	string	SIZE (0..255)	SnmpTaglist
AdminString	string	SIZE (0..255)	SnmpAdminString

OM Data Type	XML-Schema data type	Permitted Values	SNMP Mapping
RowStatus	int		RowStatus
TimeStamp	unsignedInt		TimeStamp
duration	unsignedInt	0..2147483647	TimeInterval
StorageType	int		StorageType
InetAddressPrefixLength	unsignedInt	0..2040	Unsigned32
InetPortNumber	unsignedInt	0..65535	Unsigned32
DocsisQosVersion	int		DocsisQosVersion [RFC 4546]
DocsisUpstreamType	int		DocsisUpstreamType [RFC 4546]
DocsEqualizerData	hexBinary		DocsEqualizerData [RFC 4546]
TenthdBmV	int		TenthdBmV [RFC 4546]
TenthdB	int		TenthdB [RFC 4546]

6.5.7 Derived Data Types (YANG Mapping)

Table 6-6 represents the mapping between the CCAP derived data types and their equivalent representation in YANG. The permitted values for the data types are indicated in terms of value ranges and string length when applicable. The UML Derived Data Type column includes the data types to map to YANG, using the appropriate type in YANG. The YANG Derived Data Type Mapping references YANG data types defined in [RFC 6021] or as described below.

Table 6-6 - Derived Data Types

UML Derived Data Type	YANG Derived Data Type Mapping	Permitted Values
Counter32	counter32	
Counter64	counter64	
Gauge32	gauge32	
TimeStamp	timestamp	
MacAddress	mac-address	e.g., 01:23:45:67:89:ab
InetPortNumber	port-number	0..65535
IPAddress	ip-address	IPv4 or IPv6 Address
IPv4Address	ipv4-address	IPv4 Address
IPv6Address	ipv6-address	IPv6 Address
InetAddressPrefixLength	address-prefix-len-type	0..2040
InetIpv4Prefix	ipv4-prefix	IPv4 Address "/" IPv4 Prefix Length
InetIpv6Prefix	ipv6-prefix	IPv6 Address "/" IPv6 Prefix Length
Uri	uri	
TagList	snmp-tag-list-type	String(SIZE(0..255))
AdminState	admin-state-type	other(1), up(2), down(3), testing(4)
DateTime	date-and-time	

6.6 Remote PHY Common Data Type Definitions

There are no additional data types created specifically to support the Remote PHY Information Models.

7 CONFIGURATION MANAGEMENT

In the Remote PHY architecture, the Remote PHY Device (RPD) has very minimal local configuration (e.g., certificates, passwords) and the majority of its operational configuration is provided during RPD initialization by the CCAP Core(s) via the control plane. Thus, operator configuration of the RPD is performed via configuration of the CCAP Core(s).

The R-PHY model supports configuration by a Primary CCAP Core and 0 or more Auxiliary CCAP Cores. This document will use the term CCAP Core to refer to either one.

7.1 RPD Configuration Theory of Operation

The CCAP controls its connected RPDs. A single CCAP serves as the single point of configuration for a set of resources (e.g., RF ports and channels) on a given RPD. The CCAP processes its configuration, which can include references to RPDs. Once the RPD bootstrap process has been completed, the CCAP translates applicable physical layer configuration to GCP objects. Then, the CCAP uses GCP to communicate these configuration objects to the relevant RPDs.

7.2 CCAP Configuration and Transport Protocol Requirements

7.2.1 Configuration Object Datastore

The CCAP supporting the Remote PHY architecture MUST implement the standard configuration objects defined by this specification.

The CCAP supporting the Remote PHY architecture MUST implement the standard configuration objects defined by [CCAP-OSSIv3.1], except where specified differently in this specification.

7.2.2 Dynamic Management of RPDs

When the downloaded XML-based configuration file contains information on a new RPD, the CCAP MUST provide the RPD configuration information via GCP to the new RPD.

When the downloaded XML-based configuration file contains information which modifies the configuration of a given RPD, the CCAP MUST utilize GCP to modify the RPD's configuration. The CCAP SHOULD do this in such a way as to minimize the impact of the change on other unchanged channels, ports, and functions on the RPD.

7.3 UML Configuration Object Model

7.3.1 CCAP UML Configuration Object Model Overview

For DOCSIS 3.0, 3.1, and DPoE, the CCAP UML configuration object model, as well as the schemas based on that object model, was divided into eight distinct groupings:

- CCAP: The Ccap object is the container of all CCAP configuration objects.
- Chassis: Consists of objects for configuring the hardware components of the CCAP.
- Video: Consists of those objects that are related to the EQAM functions of the CCAP, including ERM, encryption and decryption objects.
- DOCSIS: Consists of the DOCSIS configuration objects that are needed for configuring DOCSIS MAC Domains and services such as DSG.
- Network: Consists of objects related to configuring the core services for things like integrated servers, access lists, Syslog, HTTP, FTP, SSH, and other related network services.
- Interfaces: Consists of the objects needed to configure interfaces within the CCAP.
- Management: Consists of objects used to configure SNMP and Fault Management for the CCAP.
- EPON: Consists of the objects that are related to the DPoE configuration of the CCAP.

This specification extends that model to include an RPD grouping:

- RPD: Consists of objects that are related to the configuration of RPDs managed by the CCAP.

The CCAP supports the RPD-related configuration objects defined in the following sections via implementation of the CCAP XSD.

The CCAP configuration object model described in [CCAP-OSSIv3.1] has been modified in this specification for the Remote PHY architecture; those changes are described here. Objects not defined here are unchanged and detailed in [CCAP-OSSIv3.1].

7.3.1.1 Default Values and Mandatory Configuration of Attributes in the Configuration Object Model

In the configuration object model attribute tables in the following sections, a default value is defined in the Default table column for some object attributes. In cases where a default value is defined for an element, the CCAP will use the specified default value if the XML configuration file does not include the attribute.

In cases where the Default column reads "vendor-specific", the CCAP provides a default value of the vendor's choosing for the attribute in the implementation. In cases where the vendor is defining the default value, the operator need not include these attributes in the XML configuration file.

Attributes explicitly required in the XML configuration file are marked "Yes" in the Required Attribute column; these attributes do not have a default value. In these cases the operator needs to provide a value for these attributes in the XML configuration file when an object containing those attributes is being configured. In cases where the Required Attribute column reads "No", either a default value is provided in the table or the CCAP will provide a vendor-specific value.

7.3.1.2 Enumeration Values in the Configuration Object Model

In the configuration object model attribute tables in the following sections, enumerated lists are all intended to begin at a value of "1"; in most cases, the first value will be other ("other(1)"). Since this specification borrows objects from existing MIBs, there will be cases where the enumeration values specified here do not match those of the MIB on which the object attribute was based. CCAP vendors are expected to properly translate values provided in the XML configuration file into the correct values needed for SNMP reporting via the standard MIB objects.

Note that integers are specified for each enumeration in the UML configuration object model. When the UML is translated into other formats (XSD, YANG, SNMP, etc.), the enumeration labels and/or integers are included in these outputs as appropriate. For XSD and YANG, enumeration labels will be included.

7.3.1.3 Use of Interface Names in Configuration

Several configuration objects defined in this specification are identified with keys in the form of a text string name. In general, these configuration objects are modeled after interfaces that have equivalent representation in SNMP (ifTable). While this specification does not impose formal requirements on the format of interface names, CCAP vendors are expected to implement consistent conventions for assigning textual names to interfaces and disclose the rules on which such conventions are based. The CCAP typically rejects a configuration that includes an interface name that does not follow the vendor's naming conventions.

7.3.1.4 Unconstrained Strings in the Configuration Object Model

For object attributes with a data type of String, there are cases where this specification does not provide a length constraint. For these attributes, the CCAP can impose a vendor-specific length constraint. If a value in the XML configuration file exceeds this vendor-specific length constraint, the CCAP typically truncates the text string to that limit and logs an error.

7.3.2 Vendor-Specific Extensions

A CCAP is expected to implement vendor-proprietary configuration objects beyond those defined in this specification. Standard objects are those that have been defined in the configuration UML object model, defined in the following sections. Vendor-proprietary configuration objects consist of both new configuration objects not

represented in the CCAP configuration UML object model and new or modified attributes of configuration objects that exist in the CCAP configuration UML object model.

The CCAP's configuration object model can be extended via the creation of vendor-proprietary XSD schemas and/or vendor-proprietary YANG modules. A valid approach to vendor extensions is to perform extensions solely in XML schema utilizing the extension points in the standard schema (Additional details are expected in a future version of the specification.) in conjunction with a vendor-defined schema. Vendor extensions can also be performed in YANG. A CCAP that supports vendor extension in YANG also supports configuration via an XML configuration file based on an XSD schema that is the result of the conversion of the standard YANG module with extensions.

7.4 CCAP Configuration Objects

The CCAP configuration object model has been modified for the Remote PHY architecture; those changes are described in the following subsections. Objects not defined here are unchanged and are detailed in [CCAP-OSSIv3.1].

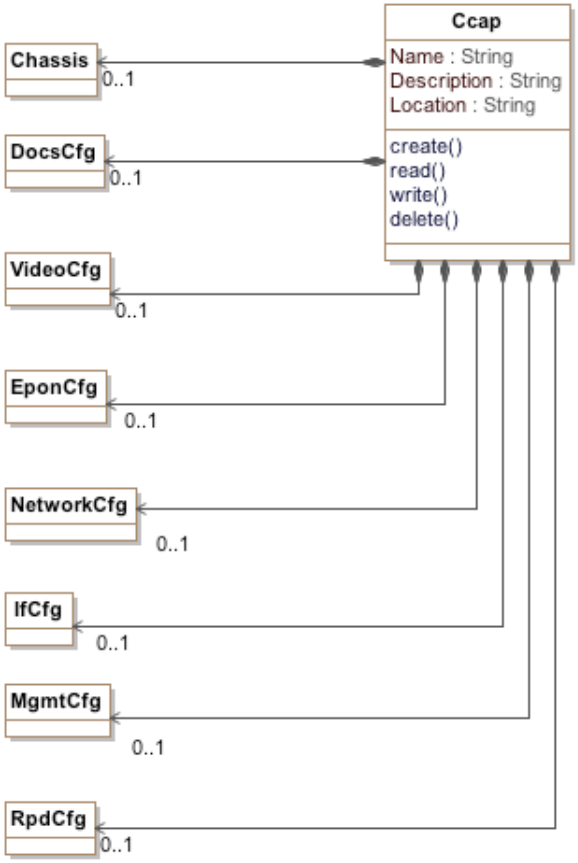

Figure 7-1 - CCAP Configuration Objects

7.4.1 Ccap Object

The Ccap object serves as the root of the CCAP configuration data. It is largely defined in [CCAP-OSSIv3.1], but is extended here via the RpdCfg object and changes beneath the Chassis object. All other objects are unmodified from [CCAP-OSSIv3.1] and are described fully there.

Table 7-1 - New Ccap Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
RpdCfg	Directed composition to RpdCfg		0..1	

7.4.1.1 RpdCfg

This configuration object is included in Figure 7-1 for reference. The RpdCfg object is defined in Section 7.4.3.

7.4.2 CCAP Chassis Objects

The Chassis configuration object model has been modified for the Remote PHY architecture; those changes are described in the following subsections. Objects not defined here are unchanged and are detailed in [CCAP-OSSIv3.1].

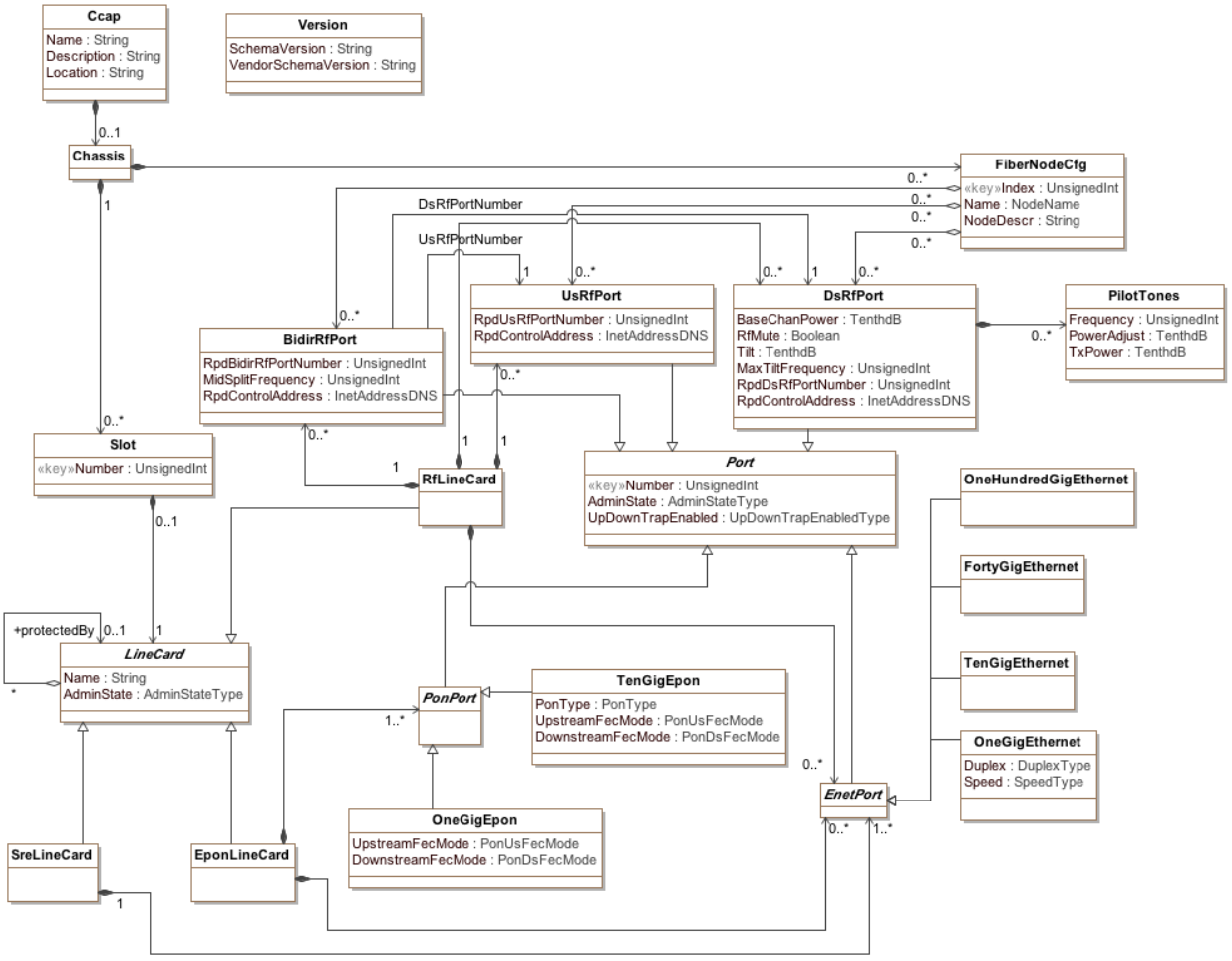

Figure 7-2 - CCAP Chassis Objects

7.4.2.1 RfLineCard

This object holds the configuration data for the RF line card in a CCAP; it is extended for R-PHY to represent the RF ports on the RPD and (optionally) the Ethernet ports on the RF Line Card used to communicate with the RPD.

For R-PHY an RfLineCard can contain zero or more logical DsRfPort and UsRfPort objects which on a CCAP Core represent the configuration of physical ports on an RPD, similar to what is traditionally defined for RfLineCards in [CCAP-OSSIv3.1]. The RfLineCard can also optionally contain zero or more BidirRfPort objects, each of which represent a physical bidirectional RF port on an RPD; each BidirRfPort references the logical DsRfPort and UsRfPort objects with which it is associated.

This definition allows the flexibility for an RfLineCard to support either 1) BidirRfPorts with associated logical DsRfPorts and UsRfPorts, 2) only DsRfPorts, 3) only UsRfPorts, or 4) both DsRfPorts and UsRfPorts. Additionally, an RfLineCard can optionally support Ethernet ports used for communication between the RfLineCard and the RPD.

A BidirRfPort instance refers to a DsRfPort and a UsRfPort instance.

The new associations for the RfLineCard are listed in Table 7-2.

Table 7-2 - New RfLineCard Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
BidirRfPort	Directed composition to BidirRfPort	1	0..*	
EnetPort	Directed composition to EnetPort	1	0..*	

7.4.2.2 Port

The Port object is an abstract class from which all physical port objects on CCAP line cards are derived. There are no Port objects instantiated *per se* in an XML instance file; only the derived physical port objects are instantiated. All physical port objects that derive from Port contain the attributes of a Port which are defined in [CCAP-OSSIv3.1].

7.4.2.3 DsRfPort

This object allows for the configuration of a physical Downstream RF port on an RfLineCard or a logical Downstream RF port whose physical port is located on an RPD. For Remote PHY, a DsRfPort object also contains the new attributes in the following table. Other attributes are unchanged from [CCAP-OSSIv3.1]. Note, however, that the Tilt and MaxTiltFrequency attributes generally only apply to a Remote PHY device deployed in a headend or hub location where the downstream signals enter the combining network and are fed to traditional analog lasers. These attributes are not configured for an RPD with a flat output, but could be configured if the RPD supports tilted output.

Table 7-3 - New DsRfPort Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
RpdDsRfPortNumber	UnsignedInt	No			
RpdControlAddress	InetAddressDNS	No			

The DsRfPort has the following new associations.

Table 7-4 - DsRfPort Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
PilotTones	Directed composition to PilotTones	1	0..*	

7.4.2.3.1 New DsRfPort Object Attributes

7.4.2.3.1.1 RpdDsRfPortNumber

This attribute identifies the physical port on the RPD that this logical DS RF port represents.

This attribute is omitted if this logical DsRfPort is associated with a physical BidirRfPort on the RPD.

7.4.2.3.1.2 RpdControlAddress

This attribute configures the IP address (v4 or v6) or FQDN of the RPD to which this logical DS RF port is associated.

This attribute is omitted if this logical DsRfPort is associated with a physical BidirRfPort on the RPD.

7.4.2.4 PilotTones

This new R-PHY object allows pilot tones to be configured on a DS RF port of an RPD. The pilot tone is commonly used on downstream RF ports for automatic gain control. Multiple pilot tones can be configured for a downstream RF port.

Table 7-5 - PilotTones Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Frequency	UnsignedInt	Yes		Hz	
PowerAdjust	UnsignedInt	Yes, see description	TenthdB		
TxPower	UnsignedInt	Yes, see description	TenthdB		

7.4.2.4.1 PilotTones Object Attributes

7.4.2.4.1.1 Frequency

This attribute is the RF frequency of this pilot tone.

7.4.2.4.1.2 PowerAdjust

This attribute represents the power gain for the pilot tone relative to the BaseChanPower for this DsRfPort. It is expressed in TenthdB.

The CCAP MUST reject a configuration of the PilotTones object that has both a PowerAdjust and a TxPower attribute configured. The CCAP MUST reject a configuration of the PilotTones object that has neither a PowerAdjust nor a TxPower attribute configured.

7.4.2.4.1.3 TxPower

This attribute represents the absolute power for the pilot tone. It is expressed in TenthdB.

7.4.2.5 UsRfPort

A UsRfPort object represents a physical Upstream RF port on an RfLineCard or a logical Upstream RF port whose physical port is located on an RPD. For Remote PHY, an UsRfPort object contains the new attributes in the following table.

Table 7-6 - New UsRfPort Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
RpdUsRfPortNumber	UnsignedInt	No			
RpdControlAddress	InetAddressDNS	No			

A UsRfPort's associations are defined in [CCAP-OSSIv3.1].

7.4.2.5.1 New UsRfPort Object Attributes

7.4.2.5.1.1 RpdUsRfPortNumber

This attribute identifies the physical port on the RPD that this logical US RF port represents.

This attribute is omitted if this logical UsRfPort is associated with a physical BidirRfPort on the RPD.

7.4.2.5.1.2 RpdControlAddress

This attribute configures the IP address (v4 or v6) or FQDN of the RPD with which this logical US RF port is associated.

This attribute is omitted if this logical UsRfPort is associated with a physical BidirRfPort on the RPD.

7.4.2.6 BidirRfPort

This new R-PHY object corresponds to a physical bidirectional RF port on an RPD. This object allows for the configuration of the association between a physical bidirectional RF port on an RPD and the logical DsRfPorts and UsRfPorts (from the CCAP's perspective) that hold the downstream and upstream configuration of that bidirectional port. The BidirRfPort is a type of the abstract class Port and inherits those common parameters. A BidirRfPort object contains the attributes in the following table.

Table 7-7 - BidirRfPort Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
RpdBidirRfPortNumber	UnsignedInt	Yes			
MidSplitFrequency	UnsignedInt	No		Hz	
RpdControlAddress	InetAddressDNS	Yes			

Table 7-8 - BidirRfPort Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
Port	Specialization of Port			
DsRfPort	Directed association to DsRfPort	1	1	DsRfPortNumber
UsRfPort	Directed association to UsRfPort	1	1	UsRfPortNumber

7.4.2.6.1 BidirRfPort Object Attributes

7.4.2.6.1.1 RpdBidiRfPortNumber

This attribute is the port number of this Bidirectional RF Port from the RPD's Bidirectional RF port number space.

7.4.2.6.1.2 MidSplitFrequency

This attribute is the RF frequency of the midsplit in the cable system connected to this Bidirectional RF Port. This attribute can be omitted if it is not configurable in the RPD.

7.4.2.6.1.3 RpdControlAddress

This attribute configures the IP address (v4 or v6) or FQDN of the RPD to which this bidirectional port is associated.

7.4.2.7 FiberNodeCfg

The FiberNodeCfg object defines the cable hybrid fiber/coax system (HFC) plant Fiber Nodes reached by RF ports on a CCAP. FiberNode attributes are defined in [CCAP-OSSIv3.1].

The FiberNodeCfg object has the following associations.

Table 7-9 - New FiberNodeCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
BidirRfPort	Directed aggregation to BidirRfPort	0..*	0..*	BidirRfPort

When using BidirRfPorts, the CCAP SHOULD reject a configuration where a FiberNodeCfg contains a DsRfPort or UsRfPort which are associated with a BidirRfPort. This is because BidirRfPorts already contain references to UsRfPorts and DsRfPorts.

7.4.3 RpdCfg Objects

The RPD configuration objects are new for the Remote PHY architecture.

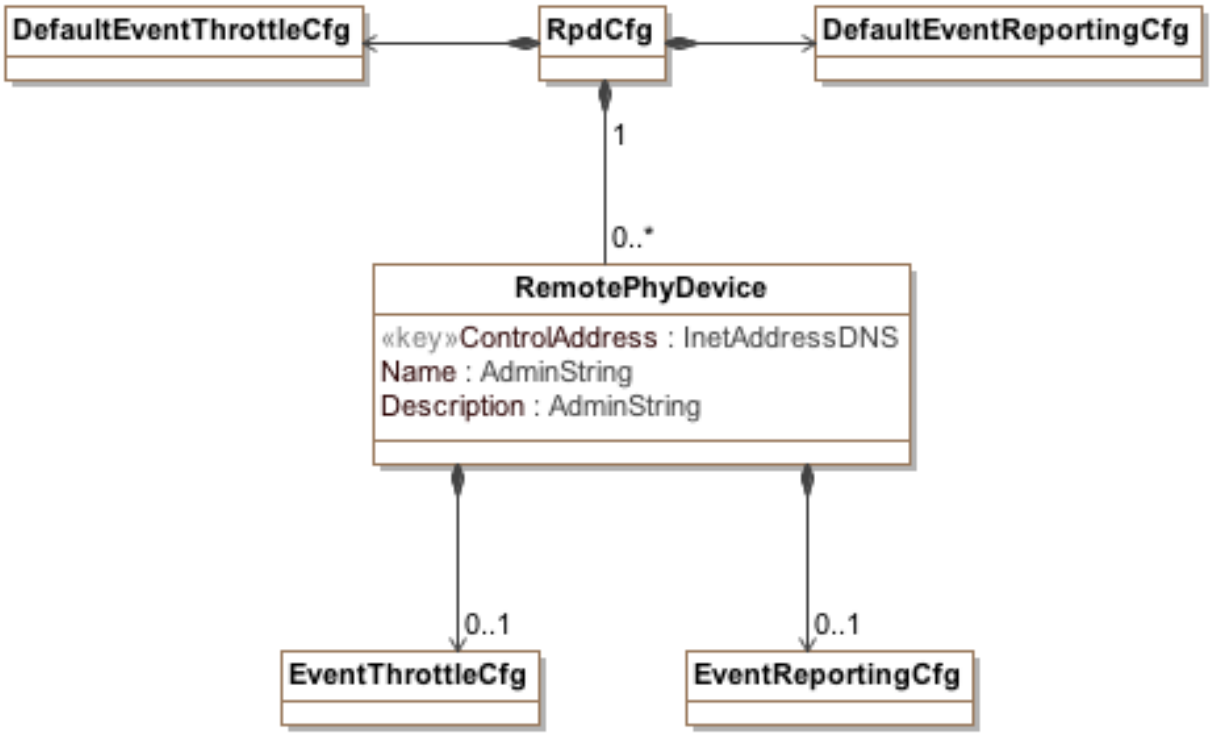

Figure 7-3 - CCAP Rpd Objects

7.4.3.1 RpdCfg

The RpdCfg object is a container that holds RemotePhyDevice instances and has the associations shown in Table 7-10.

Table 7-10 - RpdCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
RemotePhyDevice	Directed composition to RemotePhyDevice	1	0..*	
DefaultEventThrottleCfg	Directed composition to DefaultEventThrottleCfg	1	1	
DefaultEventReportingCfg	Directed composition to DefaultEventReportingCfg	1	1	

7.4.3.2 DefaultEventThrottleCfg

This object configures the default event throttling parameters for RPDs. If an instance of RemotePhyDevice is configured without an EventThrottleCfg object, the configuration here applies. It is based on the EventThrottleCfg object defined in [CCAP-OSSIv3.1] and is used here without modification.

7.4.3.3 DefaultEventReportingCfg

This object configures the default event reporting parameters for RPDs. If an instance of RemotePhyDevice is configured without an EventReportingCfg object, the configuration here applies. It is based on the EventReportingCfg object defined in [CCAP-OSSIv3.1] and is used here without modification.

7.4.3.4 RemotePhyDevice

The RemotePhyDevice object allows the user to optionally configure attributes of the Remote PHY Devices for reporting purposes.

Table 7-11 - RemotePhyDevice Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
ControlAddress	InetAddressDNS	Yes (key)			
Name	AdminString	No			
Description	AdminString	No			

Table 7-12 - RemotePhyDevice Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
EventThrottleCfg	Directed composition to EventThrottleCfg	1	0..1	
EventReportingCfg	Directed composition to EventReportingCfg	1	0..1	

7.4.3.4.1 RemotePhyDevice Object Attributes

7.4.3.4.1.1 ControlAddress

This attribute configures the IP address (v4 or v6) or FQDN of the RemotePhyDevice.

7.4.3.4.1.2 Name

This attribute configures a short name of the RemotePhyDevice for reporting. While not a key, the Name of the RemotePhyDevice is required to be unique on this CCAP.

7.4.3.4.1.3 Description

This attribute configures an informational description of the RemotePhyDevice.

7.4.3.5 EventThrottleCfg

This object configures specific event throttling parameters for a RemotePhyDevice instance. It is defined in [CCAP-OSSIv3.1].

7.4.3.6 EventReportingCfg

This object configures specific event reporting parameters for a RemotePhyDevice instance. It is defined in [CCAP-OSSIv3.1].

7.4.4 Downstream RF Port Configuration Objects

The downstream RF port configuration object model has been modified for the Remote PHY architecture; those changes are described in the following subsections. Objects not defined here are unchanged and are detailed in [CCAP-OSSIv3.1].

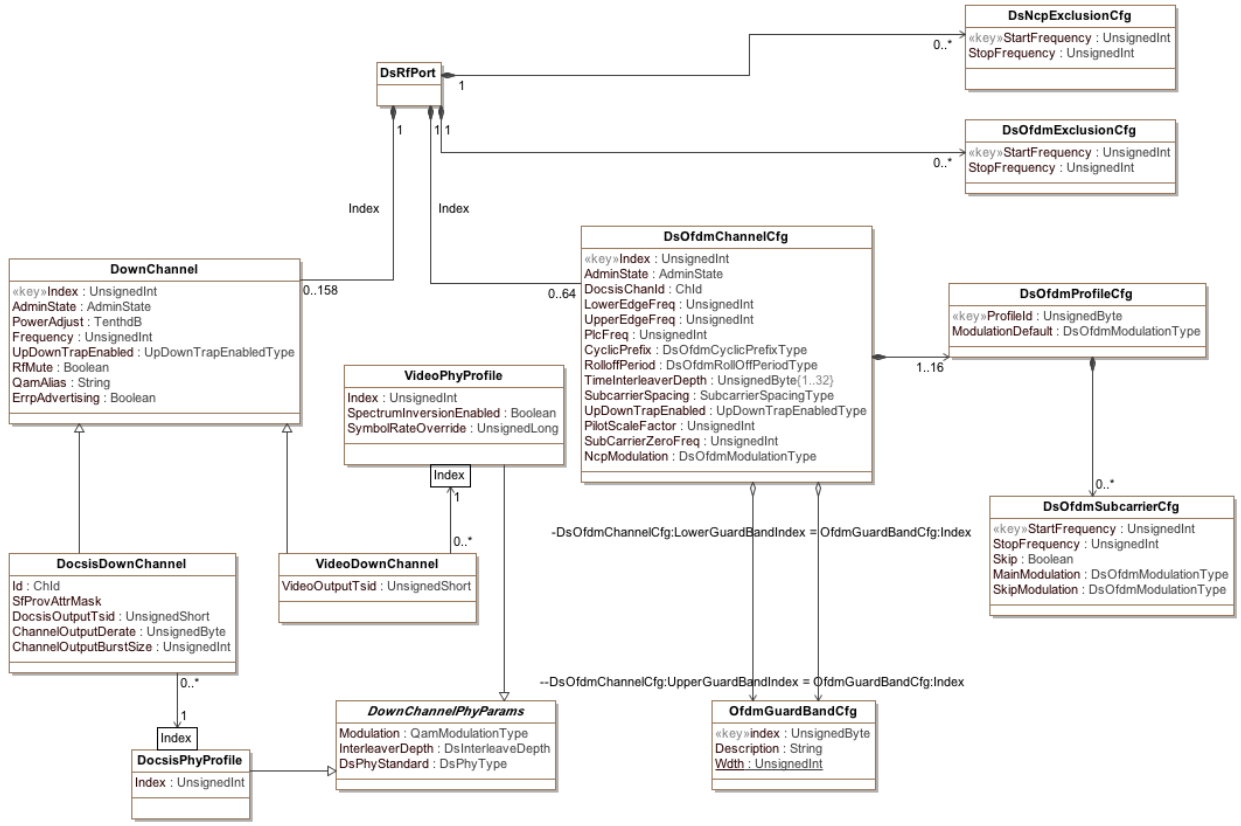

Figure 7-4 - CCAP Downstream RF Port Configuration Objects

7.4.4.1 Downstream RF Port

The DsRfPort object allows the user to configure the CCAP Downstream RF Ports elements either on an integrated CCAP or CCAP connected to a Remote PHY Device. The DsRfPort object has the following objects which are modified by this specification; all other objects and associations are as defined in [CCAP-OSSIv3.1].

7.4.4.1.1 DocsisDownChannel

The DocsisDownChannel object is a DownChannel used exclusively for DOCSIS. Two attributes have been added to the DocsisDownChannel object for Remote PHY support: ChannelOutputDerate and ChannelOutputBurstSize. Otherwise the DocsisDownChannelObject is unmodified from its definition in [CCAP-OSSIv3.1].

Table 7-13 - New DocsisDownChannel Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
ChannelOutputDerate	UnsignedByte	No	90..100	%	99%
ChannelOutputBurstSize	UnsignedInt	No		Bytes	

7.4.4.1.1.1 DocsisDownChannel Object Attributes

7.4.4.1.1.1.1 ChannelOutputDerate

The percentage of the maximum output rate for the aggregated traffic that is being sent though this Downstream interface to the Downstream channel associated with this DEPI session. Using a value lower than 100% of the Downstream channel's configured payload rate prevents the buildup of a queue delay when MPEG-TS nulls are added in the presence of jitter in the CIN.

7.4.4.1.1.1.2 ChannelOutputBurstSize

The maximum burst size for the aggregate output rate of traffic that is being sent through this Downstream interface to the Downstream channel. The default value of this object corresponds to 3 CCAP Core payload MTUs.

8 PERFORMANCE MANAGEMENT

Additional details are expected in a future version of the specification.

9 ACCOUNTING MANAGEMENT

There are no additional accounting management requirements to support MHAv2.

10 FAULT MANAGEMENT AND REPORTING REQUIREMENTS

10.1 Fault Management Requirements and Transport Protocols

This section defines requirements for remote monitoring/detection, diagnosis, reporting, and correction of problems.

10.2 Event Reporting

The CCAP MUST log events using standard mechanisms defined in section 8 of [OSSIv3.0].

The CCAP MUST support all Mandatory ("M") CMTS MIB objects that have an SNMP access type of accessible for SNMP Notifications ("Acc-FN") in Annex A of [OSSIv3.0] and Annex A of [L2VPN].

The CCAP MUST log events when loss of fan, loss of power supply, and temperature issues are detected. These events are specified in Annex A. The CCAP is expected to implement additional physical and environmental events beyond the three basic ones listed here.

10.2.1 SNMP Usage

In the DOCSIS environment, SNMP is one method is used to achieve the goals of fault management: remote detection, diagnosis, reporting, and correction of CMTS/CCAP network faults.

The CMTS/CCAP sends SNMP notifications to one or more NMSs (subject to operator imposed policy). CMTS/CCAP requirements for SNMP notifications are detailed in Section 10.2.2.1.2. The CMTS/CCAP sends events to a syslog server. The CMTS/CCAP requirements for syslog events are detailed in Section 10.2.2.1.3.

10.2.2 CCAP Core Event Notification

The CMTS/CCAP generates asynchronous events that indicate malfunction situations and notify the operator about important events. The methods for reporting events are defined below:

1. Stored in Local Log (docsDevEventTable from [RFC 4639]).
2. Reported to SNMP entities as an SNMP notification.
3. Sent as a message to a Syslog server.
4. Optionally reported to NETCONF clients as a NETCONF notification.

This specification defines the support of DOCSIS specific events (see Annex B) and IETF events. The former are normally in the form of SNMP notifications. The delivery of IETF Notifications to local log and syslog server is optional.

Event Notifications are enabled and disabled via configuration settings.

Events can be reported to Local Log, Syslog, and/or SNMP notifications based on the configuration settings defined in the EventReportingCfg object (see Section 7.4.3.6).

The CMTS and CCAP MUST support event notifications via local event logging.

The CMTS and CCAP MUST support event notifications via Syslog, including limiting/throttling, as specified in [RFC 4639].

The CMTS and CCAP MUST support event notification via SNMP traps, including limiting/throttling, as specified in [RFC 4639].

10.2.2.1 Format of Events

The subsections which follow explain in detail how the CMTS and CCAP report standard events by any of the following three mechanisms: local event logging, SNMP notification, and Syslog.

Annex B lists all DOCSIS event definitions.

10.2.2.1.1 Local Event Logging

The CCAP MUST maintain Local Log events, defined in [RFC 4639], in local non-volatile storage.

The CMTS and CCAP MAY retain events designated for local volatile storage in local non-volatile storage.

The CCAP Local Log non-volatile storage events MUST persist across reboots.

Events are identical if their EventIds are identical. For identical events occurring consecutively, the CMTS and CCAP MAY choose to store only a single event.

If the CCAP stores as a single event multiple identical events that occur consecutively, the CCAP MUST reflect the most recent event in the event description.

A CMTS MUST maintain Local Log events, defined in Annex B, in local-volatile storage or local non-volatile storage or both. A CMTS MAY retain in local non-volatile storage events designated for local volatile storage.

A CMTS MUST implement its Local Log as a cyclic buffer. The number of entries supported by the CMTS for the Local Log is vendor-specific with a minimum of ten entries. The CMTS Local Log MAY persist across reboots. The CMTS MUST provide access to the Local Log events through the docsDevEventTable [RFC 4639].

Aside from the procedures defined in this document, event recording conforms to the requirements of [RFC 4639]. Event descriptions are defined in English. A CMTS MUST implement event descriptors such that no event descriptor is longer than 255 characters, which is the maximum defined for SnmpAdminString [RFC 3411].

The EventId digit is a 32-bit unsigned integer. EventIds ranging [RFC 4639] from 0 to (2^{31} - 1) are reserved by DOCSIS. The CMTS MUST report in the docsDevEvTable [RFC 4639] the EventId as a 32-bit unsigned integer and convert the EventId from the error codes defined in Annex B to be consistent with this number format.

The CMTS MUST implement EventIds ranging from 2^{31} to (2^{32} - 1) as vendor-specific EventIds using the following format:

Bit 31 is set to indicate vendor-specific event.

Bits 30-16 contain the lower 15 bits of the vendor's SNMP enterprise number.

Bits 15-0 are used by the vendor to number events.

Section 10.2.2.1.3 describes rules to generate unique EventIds from the error code.

The [RFC 4639] docsDevEvIndex object provides relative ordering of events in the log. The creation of local-volatile and local non-volatile logs necessitates a method for synchronizing docsDevEvIndex values between the two Local Logs after reboot. A CMTS which supports local non-volatile storage MUST adhere to the rules listed below for creating local volatile and local non-volatile logs following a re-boot:

Renumber the values of docsDevEvIndex maintained in the local non-volatile log beginning with 1.

Initialize the local volatile log with the contents of the local non-volatile log.

Use the value of the last restored non-volatile docsDevEvIndex plus one as the docsDevEvIndex for the first event recorded in the new active session's local volatile log.

The CMTS MUST clear both the local volatile and local non-volatile event logs when an event log reset is initiated through an SNMP SET of the docsDevEvControl object [RFC 4639].

10.2.2.1.2 SNMP Notifications

The CCAP MUST implement the generic SNMP notifications according to Annex A.

When any event causes a generic SNMP notification occurrence in a CMTS, the CMTS MUST send notifications if throttling/limiting mechanism [RFC 4639] and other limitations [RFC 3413] do not restrict notification sending.

The CCAP MUST implement SNMP notifications defined in [DOCS-DIAG-MIB] and [DOCS-IF3-MIB].

The CCAP MUST support at least 4 SNMP trap destinations.

The CCAP MUST support the ability to filter traps individually and filter traps by priority level.

A CMTS operating in SNMP v1/v2c NmAccess mode MUST support SNMPv1 and SNMPv2c Traps as defined in [RFC 3416].

A CMTS operating in SNMP Coexistence mode MUST support SNMP notification type 'trap' and 'inform' as defined in [RFC 3416] and [RFC 3413].

The CMTS MUST send notifications for any event, if docsDevEvControl object [RFC 4639], throttling/limiting mechanism [RFC 4639] and [RFC 3413] limitations applied later do not restrict notification sending.

The CMTS MUST NOT report via SNMP notifications vendor-specific events that are not described in instructions submitted with certification testing application documentation.

10.2.2.1.3 Syslog

The CCAP MUST support at least four Syslog servers as recipients.

The CMTS and CCAP MUST support Syslog messages that communicate interface up/down events, user login/logout events, configuration changes, and access failures.

When the CCAP sends a Syslog message for a DOCSIS-defined event, the CCAP MUST send it in the following format:

```
<level>TIMESTAMP HOSTNAME CCAP[vendor]: <eventId> text vendor-specific-text
```

When the CMTS sends a syslog message for a DOCSIS-defined event, the CMTS MUST send it in the following format:

```
<level>TIMESTAMP HOSTNAME CMTS[vendor]: <eventId> text vendor-specific-text
```

Where:

- *level* is an ASCII representation of the event priority, enclosed in angle brackets, which is constructed as an OR of the default Facility (128) and event priority (0-7). The resulting level ranges between 128 and 135.
- *TIMESTAMP* and *HOSTNAME* follow the format of [RFC 3164]. The single space after *TIMESTAMP* is part of the *TIMESTAMP* field. The single space after *HOSTNAME* is part of the *HOSTNAME* field.
- *vendor* is the vendor name for the vendor-specific syslog messages or DOCSIS for the standard DOCSIS messages.
- *eventId* is an ASCII representation of the INTEGER number in decimal format, enclosed in angle brackets, which uniquely identifies the type of event. The CMTS and CCAP MUST equate the eventId with the value stored in the docsDevEvId object in docsDevEventTable. For the standard DOCSIS events this number is converted from the error code using the following rules:
    - The number is an eight-digit decimal number.
    - The first two digits (left-most) are the ASCII code for the letter in the Error code.
    - The next four digits are filled by 2 or 3 digits between the letter and the dot in the Error code with zero filling in the gap in the left side.
    - The last two digits are filled by the number after the dot in the Error code with zero filling in the gap in the left side.

  For example, event D04.2 is converted into 68000402, and Event I114.1 is converted into 73011401. This convention only uses a small portion of available number space reserved for DOCSIS (0 to 2^{31}-1). The first letter of an error code is always in upper-case. See Annex B for event definitions.
- *text* contains the textual description for the standard DOCSIS event message, as defined in Annex B.
- *vendor-specific-text* contains vendor-specific information. This field is optional.

For example, the syslog event for the event D04.2, "ToD Response received - Invalid data format", is as follows:

```
<132>CABLEMODEM[DOCSIS]: <68000402> ToD Response received - Invalid data format
```

The number 68000402 in the example is the number assigned by DOCSIS to this particular event.

The CMTS and CCAP MAY report non-DOCSIS events in the standard syslog message format [RFC 3164] rather than the DOCSIS syslog message format defined above.

When the CMTS or CCAP sends a syslog message for an event not defined in this specification, the CMTS or CCAP MAY send it according to the format and semantics of the elements defined above.

10.2.2.2 BIT Values for docsDevEvReporting [RFC 4639]

Permissible BIT values for [RFC 4639] docsDevEvReporting objects include:

1: local(0)

2: traps(1)

3: syslog(2)

4: localVolatile(8)

5: stdInterface(9)

Bit-0 means non-volatile Local Log storage and bit-8 is used for volatile Local Log storage (see Section 10.2.2.1). Bit-1 means SNMP Notifications which correspond to both SNMP Trap and SNMP Inform.

For backward compatibility with Pre-3.0 DOCSIS devices, the CMTS MUST support bit-3 in docsDevEvReporting BITS encoding for volatile Local Log storage.

DOCSIS 3.0 devices need to support bit override mechanisms during SNMP SET operations with either one-byte or two-byte BITS encoding for docsDevEvReporting for backward compatibility with Pre-3.0 DOCSIS behavior.

The CMTS MUST use the bit-3 value to set both bit-3 and bit-8 for SNMP SET operations on docsDevEvReporting using a one-byte BITS encoded value; therefore, the CMTS reports bit-3 and bit-8 with identical values for SNMP GET operations.

The CMTS MUST use the bit-8 value to set bit-3 and bit-8 for SNMP SET operations, irrespective of the bit-3 value, on docsDevEvReporting using a two or more byte BITS encoded value.

The CMTS MAY support bit-9 in docsDevEvReporting BITS encoding in accordance with [RFC 4639] definition.

A CMTS that reports an event by SNMP Notification or syslog MUST also report the event by a Local Log (volatile or non-volatile).

Combinations of docsDevEvReporting with traps(1) and/or syslog(2) bits with no Local Log bits (bit-0, bit-3 or bit-8) set are known as unacceptable combinations.

The CMTS MUST reject and report a 'Wrong Value' error for SNMPv2c/v3 PDUs or a 'Bad Value' error for SNMPv1 PDUs for any attempt to set docsDevEvReporting with unacceptable combinations.

The CMTS MUST accept any SNMP SET operation to docsDevEvReporting different than the unacceptable combinations.

The CMTS MUST ignore any undefined bits in docsDevEvReporting on SNMP SET operations and report a zero value for those bits.

Refer to Section 10.2.2.1.1 for details on Local Log requirements for the CMTS.

If CMTS supports both volatile and non-volatile storage, the CMTS MUST maintain the non-volatile storage when both non-volatile Local Log and volatile Local Log bits are set for a specific docsDevEvReporting event priority. If CMTS supports both volatile and non-volatile storage, the CMTS MAY maintain the volatile storage when both non-volatile Local Log and volatile Local Log bits are set for a specific docsDevEvReporting event priority. When both non-volatile Local Log and volatile Local Log bits are set for a specific docsDevEvReporting event priority, the CMTS MUST NOT report duplicate events in the docsDevEventTable.

10.2.2.3 Standard Events for CCAP

The CCAP MUST maintain the non-volatile storage when both non-volatile Local Log and volatile Local Log bits are set for a specific event priority, configured in the Reporting attribute of the EventReportingCfg object (see Section 7.4.3.6).

The CCAP MAY maintain the volatile storage when both non-volatile Local Log and volatile Local Log bits are set for a specific event priority.

When both non-volatile Local Log and volatile Local Log bits are set for a specific event priority, the CCAP MUST report the event as a single event in the docsDevEventTable.

Event priority levels for the CCAP use the following categories:

Emergency(1) events indicate fatal hardware or software failure that prevent normal system operation (all services are affected).

Alert(2) events indicate a major hardware or software failure that causes some service interruption (no redundancy available).

Critical(3) events indicate a major hardware or software failure that does not cause an interrupt of the normal data flow. This level of event may be also used when some redundant device was automatically activated to replace the defective device.

Error(4) events indicate that an incorrect input signal (external system error) is causing temporary or permanent interruption of the normal data flow.

Warning(5) events indicate a minor failure that does not cause any interrupt of the data flow.

Notice(6) events indicate that a specified alarm condition has been removed.

Information(7) events indicate a milestone or checkpoint in normal operation that could be of particular importance for troubleshooting.

Debug(8) events are reserved for vendor-specific events.

The reporting mechanism for each priority can be changed from the default reporting mechanism via the EventReportingCfg object defined in this specification (see Section 7.4.3.6).

10.2.2.4 Standard DOCSIS Events for CMTS

CMTSs use the same levels of the event priorities as a CM (see [CCAP-OSSIv3.1]); however, the priority definition of the events is different. Events with the priority level of 'Warning' and less, specify problems that could affect the individual user (for example, individual CM registration problem).

Every CMTS vendor may define their own set of 'Alert' events.

Priority level of 'Error' indicates problems with a group of CMs (for example CMs that share same upstream channel).

Priority level of 'Critical' indicates a problem that affects the whole cable system operation, but is not a faulty condition of the CMTS device.

Priority level of 'Emergency' is vendor-specific and indicates problems with the CMTS hardware or software, which prevents CMTS operation.

During CMTS initialization or reinitialization, the CMTS MUST support, as a minimum, the default event reporting mechanism shown in Table 10-1 or Table 10-2 or Table 10-3.

The CMTS MAY implement default reporting mechanisms above the minimum requirements listed in Table 10-1 or Table 10-2 or Table 10-3 with the exception of the 'Debug' priority level.

The reporting mechanism for each priority could be changed from the default reporting mechanism by using docsDevEvReporting object of DOCS-CABLE-DEVICE-MIB [RFC 4639].

Table 10-1 - CMTS Default Event Reporting Mechanism Versus Priority (Non-volatile Local Log Support Only)

Event Priority	Local Log Non-volatile	SNMP Notification	Syslog	Local Log Volatile
Emergency	Yes	No	No	Not Used
Alert	Yes	No	No	Not Used
Critical	Yes	Yes	Yes	Not Used
Error	Yes	Yes	Yes	Not Used
Warning	Yes	Yes	Yes	Not Used
Notice	Yes	Yes	Yes	Not Used
Informational	No	No	No	Not Used
Debug	No	No	No	Not Used

Table 10-2 - CMTS Default Event Reporting Mechanism Versus Priority (Volatile Local Log Support Only)

Event Priority	Local Log Non-volatile	SNMP Notification	Syslog	Local Log Volatile
Emergency	Not Used	No	No	Yes
Alert	Not Used	No	No	Yes
Critical	Not Used	Yes	Yes	Yes
Error	Not Used	Yes	Yes	Yes
Warning	Not Used	Yes	Yes	Yes
Notice	Not Used	Yes	Yes	Yes
Informational	Not Used	No	No	No
Debug	Not Used	No	No	No

Table 10-3 - CMTS Default Event Reporting Mechanism Versus Priority

Event Priority	Local Log Non-volatile	SNMP Notification	Syslog	Local Log Volatile
Emergency	Yes	No	No	No
Alert	Yes	No	No	No
Critical	Yes	Yes	Yes	No
Error	No	Yes	Yes	Yes
Warning	No	Yes	Yes	Yes
Notice	No	Yes	Yes	Yes
Informational	No	No	No	No
Debug	No	No	No	No

The CMTS MUST format notifications for standard DOCSIS events as specified in Annex B.

10.2.3 RPD Event Reporting

An RPD is required to generate asynchronous events that indicate malfunction situations and notify the operator about important events. This specification defines a single mechanism for the purpose of RPD event reporting. The RPD MUST report events to the Principal CCAP Core via GCP Notify messages. When the RPD is not attached to the Principal CCAP Core, the RPD MUST store event notifications in a cyclic buffer with a minimum of ten entries. The RPD allows the Principal CCAP Core to read the cyclic event report buffer upon establishment or re-establishment of the GCP connection. The RPD MUST persist undelivered event reports across reboots. Such approach is intended to enable reporting of events generated during RPD initialization or events related to GCP connectivity.

The detailed format of RPD generated event reports are expected in a future version of the specification.

The list of defined event reports generated by the RPD are expected in a future version of the specification.

The Principal Core maintains the responsibility of reporting events originating from the RPD by methods defined in section 9.2.2. of the CCAP OSSI specification or through vendor proprietary methods such as the Command Line Interface.

10.2.4 Event Priorities and Vendor-Specific Events

This specification defines events that make use of a sub-set of the Event Priority Levels. Vendor-specific events can be defined for any Event Priority Level. Table 10-4 summarizes those considerations.

A CMTS and CCAP MUST assign DOCSIS and vendor specific events as indicated in Table 10-4.

Table 10-4 - Event Priorities Assignment

Event Priority	CMTS and CCAP Event Assignment
Emergency	Vendor-specific
Alert	CMTS and CCAP and Vendor-specific (optional*)
Critical	CMTS and CCAP and Vendor-specific (optional*)
Error	CMTS and CCAP and Vendor-specific (optional*)
Warning	CMTS and CCAP and Vendor-specific (optional*)
Notice	CMTS and CCAP and Vendor-specific (optional*)
Information	CMTS and CCAP and Vendor-specific (optional*)
Debug	Vendor-specific
* Vendor-specific optional event definitions are recommended only where the CCAP allows for sufficient storage of such events.	

10.2.5 NETCONF Notifications

NETCONF Notifications is an optional mechanism that provides an asynchronous notification message service built on top of the base NETCONF protocol. The mechanism is based on the concept of clients subscribing to events belonging to named event streams. Clients can associate filter parameters with the subscriptions to receive a defined subset of all events belonging to a stream.

Notification replay is an integral part of the NETCONF Notifications framework. It provides the ability for clients to request sending (or resending) recently generated notifications based on a specific start and an optional stop time. If no stop time is provided, the notification stream will continue until the subscription is terminated.

The CCAP MAY implement NETCONF Notifications towards OSS.

If the CCAP implements NETCONF Notifications towards OSS, the CCAP MUST use the YANG module specified for this purpose in [CCAP-EVENTS-YANG].

10.2.6 Trap and Syslog Throttling, Limiting and Inhibiting

A CMTS MUST support SNMP TRAP/INFORM and syslog throttling and limiting as described in DOCS-CABLE-DEVICE-MIB [RFC 4639], regardless of SNMP mode.

10.2.7 Non-SNMP Fault Management Protocols

The OSS can use a variety of tools and techniques to examine faults at multiple layers. For the IP layer, useful non-SNMP based tools include ping (ICMP Echo and Echo Reply), and trace route (UDP and various ICMP Destination Unreachable flavors). The CMTS MUST support IP end-station generation of ICMP error messages and processing of all ICMP messages.

Syslog requirements are defined in Section 10.2.2.1.3.

10.3 Fault Management UML Object Model

10.3.1 Event Notification Objects

The objects for CCAP Event Notification are derived from the docsDevEventTable in [RFC 4639] and are used without modification.

11 SNMP AND MIB REQUIREMENTS

The SNMP MIB requirements listed below are subject to further review of applicability to RPD categories.

For example, the requirements for RPD located in Un-secured locations such as Fiber Node within HFC plant may be significantly different from RPD located in Secured locations.

Most CCAP MIB objects are used in a read-only mode for status and performance monitoring, and few RPD MIB objects are required to be more than read-only. The CCAP requires a very small set of read-create or read-write MIB objects used by operators for operational control, automation or testing tasks, but since the RPD is bootstrapped and configured via the CCAP, it largely does not.

The RPD MAY augment the required MIBs with objects from other standard or vendor-specific MIBs where appropriate. The RPD MUST implement the MIB requirements in accordance with this specification regardless of the value of an IETF MIB object's status (e.g., deprecated or optional).

If not required by this specification, deprecated objects are optional. If an RPD implements a deprecated MIB object, the RPD MUST implement the MIB object correctly according to the MIB definition.

If an RPD does not implement a deprecated MIB object, the following conditions MUST be met:

- The RPD MUST NOT instantiate the deprecated MIB object.
- The RPD MUST respond with the appropriate error/exception condition, such as noSuchObject for SNMPv2c, when an attempt to access the deprecated MIB object is made.

If not required by this specification, additional objects are optional. If an RPD implements any additional MIB objects, the RPD MUST implement the MIB object correctly according to the MIB definition.

If an RPD does not implement one or more additional objects, the following conditions MUST be met:

- The RPD MUST NOT instantiate the additional MIB object or objects.
- The RPD MUST respond with the appropriate error/exception condition, such as noSuchObject for SNMPv2c, when an attempt to access the non-existent additional MIB object is made.

If not required by this specification, obsolete objects are optional. If an RPD implements an obsolete MIB object, the RPD MUST implement the MIB object correctly according to the MIB definition.

If an RPD does not implement an obsolete MIB object, the following conditions MUST be met:

- The RPD MUST NOT instantiate the obsolete MIB object.
- The RPD MUST respond with the appropriate error/exception condition, such as noSuchObject for SNMPv2c, when an attempt to access the obsolete MIB object is made.

Objects which are not supported by this specification are not implemented by an agent.

- The RPD MUST NOT instantiate not-supported MIB objects.
- The RPD MUST respond with the appropriate error/exception condition, such as noSuchObject for SNMPv2c, when an attempt to access a not-supported MIB object is made.

11.1 Protocol and Agent Requirements

The RPD MUST support the SNMPv1 and SNMPv2c protocol.

The RPD MAY support the SNMPv3 protocol.

The RPD MUST support at least 10 SNMP Community strings with controlled access via access lists.

The IETF SNMP-related RFCs listed in Table 11-1 are supported by the RPD.

Table 11-1 - IETF SNMP-related RFCs

[RFC 3410]	Introduction and Applicability Statements for Internet Standard Management Framework
[RFC 3411]	An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks
[RFC 3412]	Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)
[RFC 3413]	Simple Network Management Protocol (SNMP) Applications
[RFC 3414]	User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)
[RFC 3415]	View-based Access Control Model (VACM) for the simple Network Management Protocol (SNMP)
[RFC 3416]	Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)
[RFC 3417]	Transport Mappings for the Simple Network Management Protocol (SNMP)
[RFC 3418]	Management Information Base for the Simple Network Management Protocol (SNMP)
[RFC 3419]	Textual Conventions for Transport Addresses
[RFC 3584]	Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework
[RFC 1901]	Introduction to Community-based SNMPv2 (Informational)
[RFC 1157]	A Simple Network Management Protocol

For support of SMIv2, Table 11-2 lists the IETF SNMP-related RFCs which are supported by the RPD.

Table 11-2 - SMIv2 IETF SNMP-related RFCs

[RFC 2578]	Structure of Management Information Version 2 (SMIv2)
[RFC 2579]	Textual Conventions for SMIv2
[RFC 2580]	Conformance Statements for SMIv2

For support of Diffie-Helman Key exchange for the User Based Security Model, Table 11-3 lists the IETF SNMP-related RFC which is optionally supported by the RPD.

Table 11-3 - Diffie-Helman IETF SNMP-related RFC

[RFC 2786]	Diffie-Helman USM Key Management Information Base and Textual Convention

11.1.1 RPD SNMP Modes of Operation

RPD SNMP Coexistence Mode is subject to the following requirements and limitations:

- The RPD MUST process SNMP v1/v2c Packets as described in [RFC 3411] through [RFC 3415] and [RFC 3584].
- If the RPD supports the SNMPv3 protocol, it MUST process SNMP v3 Packets as described in [RFC 3411] through [RFC 3415] and [RFC 3584].
- SNMP Access control is determined by the SNMP-COMMUNITY-MIB [RFC 3584], and SNMP-TARGET-MIB [RFC 3413], SNMP-VIEW-BASED-ACM-MIB [RFC 3415], and SNMP-USER-BASED-SM-MIB [RFC 3414].
- The RPD MUST support the SNMP-COMMUNITY-MIB [RFC 3584], which controls SNMPv1/v2c packet community string associations to a security name to select entries for access control in the SNMP-VIEW-BASED-ACM-MIB [RFC 3415].
- The RPD SHOULD support the SNMP-USER-BASED-SM-MIB [RFC 3414] and SNMP-VIEW-BASED-ACM-MIB [RFC 3415] to control SNMPv3 packets.
- The RPD MUST support SNMP Notification destinations as specified in the SNMP-TARGET-MIB and SNMP-NOTIFICATION-MIB [RFC 3413].

11.1.2 RPD SNMP Access Control Configuration

The RPD SNMP access control initial configuration is outside of the scope of this specification. If the RPD supports SNMPv3, the RPD MUST support the SNMPv3 key change mechanism defined in [RFC 3414].

Note that the SNMPv3 Initialization and KeyChange process is based on [RFC 2786], which always configures the SNMP agent with SNMPv3 HMAC-MD5-96 as the authentication protocol and CBC-DES as the privacy protocol, both specified in [RFC 3414]. Therefore, this specification does not provide a mechanism to initialize SNMPv3 using any other configuration defined in [RFC 3414].

[RFC 2786] provides a mechanism to kick start an SNMPv3 agent User-based Security Model [RFC 3414] and extensions to the same model for key change. [RFC 2786] does not define the mechanism to configure the initial key material for the kickstart process.

11.1.3 IPv6 Transport Requirements

Several transport domains were initially defined for SNMP (see [RFC 3417]). To support IPv6, [RFC 3419] adds a new set of transport domains not only for SNMP but for any application protocol.

The RPD MUST support the recommendations of [RFC 3419] to support SNMP over IPv6.

11.2 CableLabs MIBs

Table 11-4 - CableLabs MIBs

Reference	MIB Module
[DOCS-RPHY-MIB]	Placeholder
[RFC 2669]	DOCS-CABLE-DEVICE-MIB
[DOCS-IFEXT2-MIB]	DOCS-IFEXT2-MIB
[DOCS-IF3-MIB]	DOCS-IF3-MIB
[CLAB-TOPO-MIB]	CLAB-TOPO-MIB

The RPD MUST support read-only access for all Mandatory ("M") MIB objects that have an SNMP access type of read-only ("RO") in Annex A. Some objects listed as read-only in Annex A have an SNMP access type of read-write or read-create in their MIB definition. The RPD MAY additionally implement these access types, but read-only is the only access type required for support by the RPD.

11.3 IETF MIBs

Table 11-5 - IETF RFC MIBs

Reference	MIB Module
[RFC 2573]	SNMP Applications
[RFC 2786]	Diffie-Helman USM Key MIB Module: SNMP-USM-DH-OBJECTS-MIB
[RFC 2790]	Host Resources MIB Module: HOST-RESOURCES-MIB
[RFC 2863]	Interfaces Group MIB Module: IF-MIB
[RFC 3410] [RFC 3411] [RFC 3412] [RFC 3413] [RFC 3414] [RFC 3415]	SNMPv3 MIB Modules: SNMP-FRAMEWORK-MIB, SNMP-MPD-MIB, SNMP-NOTIFICATION-MIB, SNMP-TARGET-MIB, SNMP-USER-BASED-SM-MIB, SNMP-VIEW-BASED-ACM-MIB,
[RFC 3418]	SNMPv2 MIB Module: SNMPv2-MIB

Reference	MIB Module
[RFC 3433]	Entity Sensor MIB Module: ENTITY-SENSOR-MIB
[RFC 3584]	SNMP-COMMUNITY-MIB
[RFC 3635]	Ethernet Interface MIB Module: EtherLike-MIB
[RFC 4022]	Transmission Control Protocol MIB Module: TCP-MIB
[RFC 4113]	User Datagram Protocol MIB Module: UDP-MIB
[RFC 4133]	Entity MIB Module: ENTITY-MIB
[RFC 4188]	Bridge MIB Module: BRIDGE-MIB
[RFC 4293]	Internet Protocol MIB Module: IP-MIB
[RFC 4639]	DOCSIS Device MIB Module: DOCS-CABLE-DEVICE-MIB
[RFC 5601]	PW-STD-MIB

The DOCSIS OSSI 3.1 specifications have priority over the IETF MIBs and all objects. Though deprecated or optional in the IETF MIB, the object can be required by this specification as mandatory.

11.4 Specific MIB Object Implementation Requirements

11.4.1 Treatment and Interpretation of MIB Counters

Octet and packet counters implemented as counter32 and counter64 MIB objects are monotonically increasing positive integers with no specific initial value and a maximum value based on the counter size that will roll over to zero when it is exceeded. In particular, counters are defined such that the only meaningful value is the difference between counter values as seen over a sequence of counter polls. However, there are two situations that can cause this consistent monotonically increasing behavior to change: 1) resetting the counter due to a system or interface reinitialization or 2) a rollover of the counter when it reaches its maximum value of 2^{32}-1 or 2^{64}-1. In these situations, it needs to be clear what the expected behavior of the counters should be.

Case 1: The state of an interface changes, resulting in an "interface counter discontinuity" as defined in [RFC 2863].

In the case where the state of an interface within the RPD changes resulting in an "interface counter discontinuity" [RFC 2863], the RPD value of the ifXTable.ifXEntry.ifCounterDiscontinuityTime for the affected interface MUST be set to the current value of sysUpTime and ALL counters for the affected interface set to ZERO. When setting the ifAdminStatus of the affected interface to down(2), the RPD MUST NOT consider this as an interface reset.

Case 2: SNMP Agent Reset.

An SNMP Agent Reset is defined as the reinitialization of the SNMP Agent software caused by a device reboot or device reset initiated through SNMP.

In the case of an SNMP Agent Reset within the RPD, the RPD MUST:

- Set the value of sysUpTime to zero (0)
- Set all interface ifCounterDiscontinuityTime values to zero (0)
- Set all interface counters to zero (0)
- Set all other counters maintained by the RPD SNMP Agent to zero (0).

Case 3: Counter Rollover.

When a counter32 object within the RPD reaches its maximum value of 4,294,967,295, the next value MUST be zero. When a counter64 object within the RPD reaches its maximum value of 18,446,744,073,709,551,615, the next value MUST be zero.

NOTE: Unless an RPD vendor provides a means outside of SNMP to preset a counter64 or counter32 object to an arbitrary value, it will not be possible to test any rollover scenarios for counter64 objects (and many counter32 objects as well). This is because it is not possible for these counters to roll over during the service life of the device (see discussion in section 3.1.6 of [RFC 2863]).

11.4.2 Requirements for DOCSIS Device MIB [RFC 4639]

The RPD MUST implement parts of [RFC 4639] as specified in Annex A.

NOTE: [RFC 4639] includes Compliance requirements for DIFFSERV-MIB [RFC 3289] to support IPv6 filtering as a replacement for the deprecated docsDevFilterIpTable. For backwards compatibility, this specification has requirements for docsDevFilterIpTable. IPv6 filtering requirements are specified in Annex A. This specification does not define requirements for [RFC 3289].

11.4.3 Requirements for SNMPv2 MIB [RFC 3418]

11.4.3.1 SNMPv2-MIB System Group Requirements

The RPD MUST implement the System Group of [RFC 3418].

The RPD MUST implement the sysDescr object. For the RPD, the format and content of the information in sysDescr is vendor-dependent.

11.4.3.2 SNMPv2-MIB SNMP Group Requirements

This group provides SNMP protocol statistics and protocol errors counters.

The RPD MUST implement The SNMP Group from [RFC 3418].

11.4.4 Requirements for Interfaces Group MIB [RFC 2863]

The RPD MUST implement the interface MIB [RFC 2863].

The ifType object associated with a DOCSIS interface can have the following enumerated values:

- CATV downstream channel: docsCableDownstream (128)
- CATV downstream OFDM interface: docsOfdmDownstream (277)
- CATV upstream interface: docsCableUpStream (129)
- CATV upstream OFDMA interface: docsOfdmaUpstream (278)
- CATV logical upstream channel: docsCableUpstreamChannel (205)
- CATV upstream RF port: docsCableUpstreamRfPort (256)
- CATV downstream RF port: docsCableDownstreamRfPort (257)
- CATV bi-directional RF port: docsCableBidirRfPort (XYZ)

  XYZ - Additional details are expected in a future version of the document.

The following statements define the RPD interface-numbering scheme requirements:

The RPD MUST implement an instance of ifEntry for each downstream channel, upstream interface, logical upstream channel, and any other interface type that exists in the RPD.

The RPD MUST implement a row entry in the ifTable for each Bidirectional RF Port in the RPD chassis. A Bidirectional RF Port is typically associated with a single F-connector or a single MCX-75 connector. The RPD MUST implement an ifType value of XYZ in the ifTable row entry for each Bidirectional RF Port.

The RPD MUST implement a row entry in the ifTable for each Downstream RF Port in the RPD chassis. A Downstream RF Port is typically associated with a single F-connector or single MCX-75 connector. The RPD MUST implement an ifType value of 257 in the ifTable row entry for each Downstream RF Port.

When an instance of VideoDownChannel is created on a given Downstream or Bidirectional RF Port, the RPD MUST create an ifTable entry with an ifType value of 214 (QAM). For replicated QAMs, an ifTable entry will be created for every instance of a QAM on a given Downstream or Bidirectional RF Port, regardless of whether the QAM has been replicated.

When an instance of DocsisDownChannel is created on a given Downstream or Bidirectional RF Port, the RPD MUST create an ifTable entry with an ifType value of 128 (docsCableDownstream).

When an instance of DOCSIS OFDMDownstreamChannel is created on a given Downstream or Bidirectional RF Port, the RPD MUST create an ifTable entry with an ifType value of 278 (docsOfdmDownstream).

In the absence of user configuration, the RPD MAY automatically instantiate ifTable entries for VideoDownChannel objects and/or DocsisDownChannel objects.

The RPD MUST implement a row entry in the ifTable for each Upstream RF Port in the RPD chassis. An Upstream RF Port is typically associated with a single F-connector or a single MCX-75 connector. The RPD MUST implement an ifType value of 256 in the ifTable row entry for each Upstream RF Port.

When an instance of DOCSIS UpstreamPhysicalChannel is created on a given Upstream or Bidirectional RF Port, the RPD MUST automatically create one or more corresponding instances of an UpstreamLogicalChannel.

When an instance of DOCSIS UpstreamPhysicalChannel is created on a given Upstream or Bidirectional RF Port, the RPD MUST create an ifTable entry with an ifType value of 129 (docsCableUpstream).

When an instance of DOCSIS OFDMAUpstreamChannel is created on a given Upstream or Bidirectional RF Port, the RPD MUST create an ifTable entry with an ifType value of 278 (docsOfdmaUpstream).

When an instance of DOCSIS UpstreamLogicalChannel is created, the RPD MUST create an ifTable entry with an ifType value of 205 (docsCableUpstreamChannel).

In the absence of user configuration, the RPD MAY automatically instantiate DOCSIS UpstreamPhysicalChannels of ifType 129 for each physical Upstream RF port on a ULC.

For each loopback interface that is defined in the system, the RPD MUST represent that interface with an ifTable entry with an ifType value of 24, per [RFC 2863].

For each row entry created in the ifTable, the RPD MUST create a corresponding row entry in the ifXTable.

The RPD SHOULD maintain the same ifIndex value for configured interfaces across reboots if there have been no configuration changes. The interfaces to be persisted across reboots include those interfaces specified in the RPD configuration UML object model.

11.4.4.1 ifAdminStatus, ifOperStatus and Traffic

The ifAdminStatus of RF channel interfaces (i.e., those of type 128, 129, 277, and 278) is based on configuration by the CCAP Core. The RPD MUST reject attempts to set ifAdminStatus of any interface via SNMP. The RPD MUST report the value of ifAdminStatus as configured by the CCAP Core.

The RPD reports ifOperStatus according to the last CCAP-configured value of ifAdminStatus, or as 'down' for locally detected interface failure.

In the event that there is no connectivity between the CCAP Core and RPD, the RPD MUST determine an appropriate value for ifOperStatus autonomously.

The RPD MUST set the ifOperStatus on non-RF channel interfaces based on local status.

11.4.4.2 SNMP Notification Control Requirements

If a multi-layer interface model is present in the device, each sub-layer for which there is an entry in the ifTable can generate linkUp/Down traps. Since interface state changes would tend to propagate through the interface stack (from

top to bottom, or bottom to top), it is likely that several traps would be generated for each linkUp/Down occurrence. The ifLinkUpDownTrapEnable object allows managers to control SNMP notification generation, and configure only the interface sub-layers of interest.

The RPD MUST NOT transmit link notifications for RF channel interfaces.

The RPD reports link notifications for other interfaces as configured by the CCAP Core. At startup, the RPD MUST initialize the value of ifLinkUpDownTrapEnable to disabled(2). Thereafter, the RPD MUST report the value of ifLinkUpDownTrapEnable as configured by the CCAP Core. The RPD MUST reject attempts to write via SNMP the value of ifLinkUpDownTrapEnable.

11.4.4.3 ifTable and ifXTable Counters

The RPD MUST implement the ifTable and ifXTable [RFC 2863] Counter32 and Counter64 MIB objects as defined for each interface in Table 11-7 and Table 11-8.

11.4.4.4 ifSpeed and ifHighSpeed

The RPD MUST report in ifSpeed and ifHighSpeed MIB objects the current configured speed of the interface as stated in [RFC 2863]. For RPD VideoDownChannels and DocsisDownChannels, this is the symbol rate multiplied by the number of bits per symbol. For RF Upstream, this is the raw bandwidth in bits per second of this interface, of the highest speed modulation profile that is defined. This is the symbol rate multiplied with the number of bits per symbol for this modulation profile. For OFDM channels refer to [CCAP-OSSIv3.1].

11.4.4.5 ifStack Table

Shown below is an example of how the ifStack table might look for RF interfaces on the RPD. The values used for the ifIndexes are for example purposes only. The relationships are consistent with those defined in [CCAP-OSSIv3.1] but also add the Bidirectional RF Port, which is an RPD-only concept.

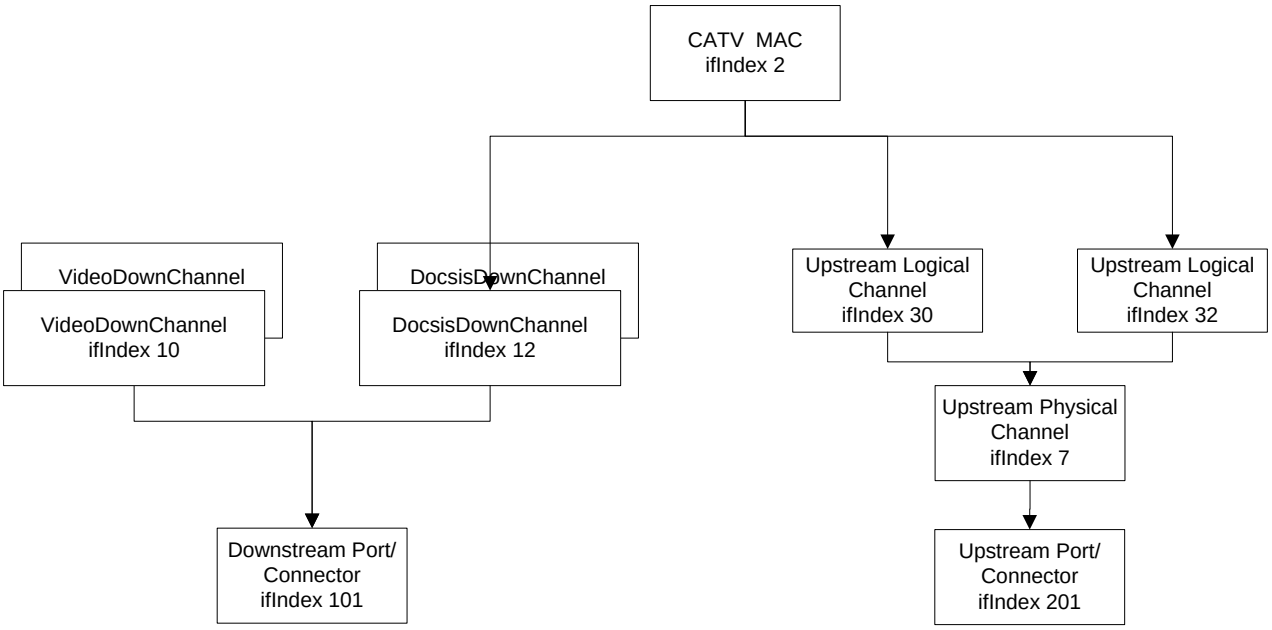

Figure 11-1 - ifStack Table for RPD RF Interfaces

Table 11-6 - RPD ifStack Table Representation

ifName	ifIndex	ifStackHigherLayer	ifStackLowerLayer
CatvMac	2	0	12
CatvMac	2	0	30

ifName	ifIndex	ifStackHigherLayer	ifStackLowerLayer
CatvMac	2	0	32
UpstreamLogicalChannel	30	2	7
UpstreamLogicalChannel	32	2	7
UpstreamPhysicalChannel	7	30	201
UpstreamPhysicalChannel	7	32	201
DocsisDownChannel	12	2	101
VideoDownChannel	10	0	101
DownstreamRfPort	101	10	0
DownstreamRfPort	101	12	0
UpstreamRfPort	201	7	0

11.4.4.6 IF-MIB Detailed Requirements

Table 11-7, Table 11-8, and Table 11-9 detail the specific ifTable and ifXTable MIB objects and values that are expected for the interfaces on the RPD.

Table 11-7 - IfTable/IfXTable Details for Ethernet Interfaces

MIB Objects	RPD-Ethernet
IfTable	
ifIndex	(n)
ifDescr	
ifType	6
ifMtu	1522 - 2000
ifSpeed (bps) Note: Interfaces higher than 10Gbps are not shown in this MIB Object. These interface speeds are recorded in the ifXTable ifHighSpeed MIB Object.	100,000 1,000,000,000, 10,000,000,000
ifPhysAddress	MAC Address of this interface
ifAdminStatus	up(1), down(2), testing(3)
ifOperStatus	up(1), down(2), testing(3), dormant(5), notPresent(6)
ifLastChange	
ifXTable	
ifName	
ifLinkUpDownTrapEnable	
ifHighSpeed (Mbits/sec)	100, 1000, 10,000, 40,000, 100,000
ifPromiscuousMode	true(1), false(2)
ifConnectorPresent	
ifAlias	
ifCounterDiscontinuityTime	

Table 11-8 - IfTable/IfXTable for RF Interfaces

MIB Objects	RPD VideoDownChannel	RPD DocsisDownChannel	RPD-Upstream Physical Channel	RPD-Upstream Logical Channel	RPD DsRfPort	RPD UsRfPort	RPD BidirRf Port	RPD DsOfdm Channel	RPD UsOfdmaChannel
IfTable									
ifIndex	(n)	(n)	(n)	(n)	(n)	(n)	(n)	(n)	(n)
ifDescr									
ifType	214*	128	129	205	257	256	XYZ	277	278
ifMtu For RF Upstream/Downstream; the value includes the length of the MAC header.	188	1764	1764	1764	0	0	0	2030	2030
ifSpeed Refer to 11.4.4.4	DVB-C ~QAM64= 41,712,000 ~QAM256= 55,616,000 J.83 Annex B ~QAM64= 30,341,646 ~QAM256= 42,884,296	DVB-C ~QAM64= 41,712,000 ~QAM256= 55,616,000 J.83 Annex B ~QAM64= 30,341,646 ~QAM256= 42,884,296	(n)	(n)	0	0	0	(n)	(n)
ifPhysAddress	Empty-String	Empty-String	Empty-String	Empty-String	Empty-String	Empty-String	Empty-String	Empty-String	Empty-String
ifAdminStatus Refer to 11.4.4.1	up(1), down(2), testing(3)	up(1), down(2), testing(3)	up(1), down(2), testing(3)	up(1), down(2), testing(3)	up(1), down(2), testing(3)	up(1), down(2), testing(3)	up(1), down(2), testing(3)	up(1), down(2), testing(3)	up(1), down(2), testing(3)
ifOperStatus Refer to 11.4.4.1	up(1), down(2), testing(3), dormant(5), notPresent(6)	up(1), down(2), testing(3), dormant(5), notPresent(6)	up(1), down(2), testing(3), dormant(5), notPresent(6)	up(1), down(2), testing(3), dormant(5), notPresent(6)	up(1), down(2), testing(3), dormant(5), notPresent(6)	up(1), down(2), testing(3), dormant(5), notPresent(6)	up(1), down(2), testing(3), dormant(5), notPresent(6)	up(1), down(2), testing(3), dormant(5), notPresent(6)	up(1), down(2), testing(3), dormant(5), notPresent(6)
ifLastChange									
ifXTable									
ifName									
ifLinkUpDownTrapEnable Refer to 11.4.4.2									

MIB Objects	RPD VideoDownChannel	RPD DocsisDownChannel	RPD-Upstream Physical Channel	RPD-Upstream Logical Channel	RPD DsRfPort	RPD UsRfPort	RPD BidirRf Port	RPD DsOfdm Channel	RPD UsOfdmaChannel
IfTable									
ifHighSpeed Refer to 11.4.4.4	DVB-C ~QAM64=41, ~QAM256=55 J.83 Annex B ~QAM64=30, ~QAM256=42	DVB-C ~QAM64=41, ~QAM256=55 J.83 Annex B ~QAM64=30, ~QAM256=42	(n)	(n)	0	0	0	(n)	(n)
ifPromiscuousMode		false(2)	true(1), false(2)	true(1)	false(2)	false(2)	false(2)	false(2)	false(2)
ifConnectorPresent									
ifAlias									
ifCounterDiscontinuityTime									

Table 11-9 - RPD ifCounters Information

MIB Counter Objects	RPD- Video Down Channel	RPD-Docsis Down Channel	RPD-Upstream Physical Channel	RPD-Upstream Logical Channel	RPD- Ds RfPort	RPD- Us RfPort	RPD-BidirRf Port	RPD DsOfdmChannel	RPD UsOfdmaChannel
ifTable									
ifInOctets For RF Upstream/Downstream (where not zero); The count of octets from ifInUcastPkts.	N/A	N/A	Optional	Optional	N/A	N/A	N/A	N/A	Optional
ifInUcastPkts For SCQAM Upstream Logical Channel, counts upstream DOCSIS frames for non-MTC channels and upstream segments for MTC channels. For UsOfdma channels, counts upstream DOCSIS frames.	N/A	N/A	N/A	Optional	N/A	N/A	N/A	N/A	Optional
ifInDiscards	Mandatory	N/A	N/A	Optional	N/A	N/A	N/A	N/A	Optional
ifInErrors	Mandatory	N/A	N/A	Optional	N/A	N/A	N/A	N/A	Optional
ifInUnknownProtos	N/A	N/A	N/A	Optional	N/A	N/A	N/A	N/A	Optional
ifOutOctets For RF Upstream/ Downstream (where not zero); This includes MAC packets as well as data packets, and includes the length of the MAC header, this does not include any PHY overhead.	Mandatory	Mandatory	N/A	N/A	N/A	N/A	N/A	Mandatory	N/A

MIB Counter Objects	RPD- Video Down Channel	RPD-Docsis Down Channel	RPD-Upstream Physical Channel	RPD-Upstream Logical Channel	RPD- Ds RfPort	RPD- Us RfPort	RPD-BidirRf Port	RPD DsOfdmChannel	RPD UsOfdmaChannel
ifOutUcastPkts For RF Upstream/Downstream; Reports zero if implemented	N/A	O	N/A	N/A	N/A	N/A	N/A	O	N/A
ifOutDiscards	N/A	O	N/A	N/A	N/A	N/A	N/A	O	N/A
ifOutErrors	N/A	O	N/A	N/A	N/A	N/A	N/A	O	N/A
ifXTable									
ifInMulticastPkts For RF Upstream/Downstream; Reports zero if implemented.	N/A	N/A	Optional	Optional	N/A	N/A	N/A	N/A	Optional
ifInBroadcastPkts For RF Upstream/Downstream; Reports zero if implemented.	N/A	N/A	Optional	Optional	N/A	N/A	N/A	N/A	Optional
ifOutMulticastPkts For RF Upstream/Downstream; Reports zero if implemented.	N/A	O	N/A	N/A	N/A	N/A	N/A	O	N/A
ifOutBroadcastPkts For RF Upstream/Downstream; Reports zero if implemented.	N/A	O	N/A	N/A	N/A	N/A	N/A	O	N/A
IfHCInOctets The count of octets from ifInHCUcastPkts.	Optional	N/A	Optional	Optional	N/A	N/A	N/A	N/A	Optional
ifHCInUcastPkts For SCQAM Upstream Logical Channel, counts upstream DOCSIS frames for non-MTC channels and upstream segments for MTC channels. For UsOfdma channels, counts upstream DOCSIS frames.	N/A	N/A	Optional	Optional	N/A	N/A	N/A	N/A	Optional
ifHCInMulticastPkts For RF Upstream/Downstream; Reports zero if implemented.	N/A	N/A	Optional	Optional	N/A	N/A	N/A	N/A	Optional
ifHCInBroadcastPkts For RF Upstream/Downstream; Reports zero if implemented.	N/A	N/A	Optional	Optional	N/A	N/A	N/A	N/A	Optional

MIB Counter Objects	RPD- Video Down Channel	RPD-Docsis Down Channel	RPD- Upstream Physical Channel	RPD- Upstream Logical Channel	RPD- Ds RfPort	RPD- Us RfPort	RPD-BidirRf Port	RPD DsOfdmChannel	RPD UsOfdmaChannel
ifHCOutOctets For RF Upstream/Downstream (where not zero); This includes MAC packets as well as data packets, and includes the length of the MAC header, this does not include any PHY overhead.	Mandatory	M	N/A	N/A	N/A	N/A	N/A	M	N/A
ifHCOutUcastPkts For RF Upstream/Downstream; Reports zero if implemented.	N/A	O	N/A	N/A	N/A	N/A	N/A	O	N/A
ifHCOutMulticastPkts For RF Upstream/Downstream; Reports zero if implemented.	N/A	O	N/A	N/A	N/A	N/A	N/A	O	N/A
ifHCOutBroadcastPkts For RF Upstream/Downstream; Reports zero if implemented.	N/A	O	N/A	N/A	N/A	N/A	N/A	O	N/A

11.4.5 Requirements for Entity-MIB [RFC 4133]

The RPD MAY implement the ENTITY-MIB [RFC 4133].

The CCAP Core MUST implement a row entry in the entPhysicalTable for each known RPD and RPD interface.

The RPD MUST provide the unique component serial number, via the entPhysicalSerialNum object, contained within the row entry in the entPhysicalTable for the system chassis and each FRU that has a serial number in the system. Example FRUs with serial numbers include, but are not limited to, DTI cards, DLCs, ULCs, combined Upstream & Downstream line cards, and Ethernet cards.

The RPD SHOULD provide the unique component serial number, via the entPhysicalSerialNum object, contained within the row entry in the entPhysicalTable for each FRU that is a pluggable optical module such as an SFP, SFP+, QSFP, XFP, CXP.

Example FRUs that might not have serial numbers, yet are expected to be represented in the entPhysicalTable, include flash cards, fan modules, and power supply modules.

The RPD MUST implement a row entry in the entPhysicalTable for the system chassis with an entPhysicalClass value of "chassis".

The RPD MUST implement row entries in the entPhysicalTable for temperature sensors in the system with an entPhysicalClass value of "sensor".

The RPD SHOULD implement a row entry in the entPhysicalTable for each system chassis slot with an entPhysicalClass value of "container".

For each row entry created in the SNMPv2-MIB ifTable that can be mapped to an entity represented in the Entity-MIB, the RPD MUST create a corresponding row entry in the entAliasMappingTable.

The RPD MUST implement a row entry in the entAliasMappingTable for each UsRfPort, DsRfPort and BidirRfPort in the chassis.

The RPD SHOULD provide the unique component serial number, via the entPhysicalSerialNum object, contained within the row entry in the entPhysicalTable for every FRU that is capable of causing and/or generating an event, message, log, or alarm.

11.4.5.1 Guidelines for the implementation of the Entity MIB

The Entity MIB [RFC 4133] provides a physical component layer applicable to managed objects defined for DOCSIS devices. In particular for the entPhysicalTable MIB objects, not all the physical components listed need to instantiate all the object's attributes in entPhysicalTable (the Maximum Access is as defined in [RFC 4133].) Therefore, Annex A, Table A-3, columns "RPD" with value "O" (optional) need to be interpreted on a physical component basis as well as the column "Access".

The following table represents high level constraints for any instance of entPhysicalTable.

Table 11-10 - entPhysicalTable Requirements

MIB object	Value
entPhysicalIndex	n
entPhysicalDescr	Text Description
entPhysicalVendorType	Enterprise-specific OID or zeroDotZero
entPhysicalContainedIn	0..n
entPhysicalClass	Physical Class per [RFC 4133]
entPhysicalParentRelPos	-1..n per [RFC 4133]
entPhysicalName	Physical element name In case of a component mapped to an interface Index ifName can be reported, otherwise zero-length string
entPhysicalHardwareRev	Hardware revision or zero-length string

MIB object	Value
entPhysicalFirmwareRev	Firmware revision or zero-length string
entPhysicalSoftwareRev	Software revision or zero-length string
entPhysicalSerialNum	Serial Number or zero-length string
entPhysicalMfgName	Manufacturer Name or zero-length string
entPhysicalModelName	Model Name or zero-length string
entPhysicalAlias	Physical element operator defined alias In case of a component mapped to an interface Index ifAlias can be reported and implemented as read-only, otherwise zero-length string
entPhysicalAssetID	User defined Asset ID or zero-length string
entPhysicalIsFRU	'true' or 'false'
entPhysicalMfgDate	Manufacturer data or all zeros '0000000000000000'H
entPhysicalUris	URI or zero-length string

The following sections detail requirements for the RPD on specific topics where the DOCSIS requirements interact with the Entity MIB.

11.4.5.2 Entity-MIB Requirements for entLogicalTable, entLPMappingTable and entConfigChange Notification

The RPD is not required to support multiple naming scopes. Therefore, this specification has no RPD requirements for entLogicalTable and entLPMappingTable and is left for vendor-specific implementation.

In addition, this specification has no RPD requirements for the entConfigChange Notification and is left for vendor-specific implementation.

11.4.5.3 Entity-MIB RPD Requirements for entPhysicalTable

The RPD MAY provide as much information as possible for major components such as chassis, backplanes and containers or modules in the form or cards and/or field replaceable units (FRUs) when possible. Modules within a modules (or card) or other contained physical components need not be detailed.

The RPD MAY report an entry in the entPhysicalTable for the chassis component with Physical Class 'chassis'.

The RPD MAY report entries in the entPhysicalTable of Physical Class 'container' (such as slots) that contains physical Field Removable Units (FRU) normally modeled as elements of Physical Class 'module'.

The RPD MAY report temperature sensors in the form of instances in the entPhysicalTable for elements of Physical Class 'sensor' with the corresponding entPhySensorType 'celsius' value in the corresponding entPhySensorTable instance of the ENTITY-SENSOR-MIB [RFC 3433].

The [DRFI] specification defines a multi-channel RF port capability. The set of downstream channels within the same RF port is also known as a "Channel Block" (see [DRFI]).

The [MULPIv3.1] specification does not have a concrete definition of multiple upstream interfaces being part of the same RF spigot as [DRFI] does for downstream channels, but in several diagrams (e.g., [MULPIv3.1] Figure 5-5) those options are discussed. For the upstream interfaces, only the physical upstream interfaces are modeled in the Entity MIB.

A Channel Block is defined as the set of downstream interfaces (Physical Class 'port') that share the same immediate physical component of Physical Class 'module' in the containment tree (entPhysicalContainsTable).

The RPD MAY report RF port as Physical Class 'module' elements in the entPhysicalTable. The RPD MAY include the text "RF port" within the description of the SNMP object entPhysicalDescr for RF ports modeled in the entPhysicalTable.

The RPD MAY report downstream interfaces (ifType = 128 and ifType = 277), as Physical Class 'port' in the entPhysicalTable.

The RPD MAY report upstream interfaces (ifType = 129 and ifType = 278) as Physical Class 'port' in the entPhysicalTable. Upstream logical channels are not represented in the entPhysicalTable as those are subinterfaces illustrated in the ifStackTable [RFC 2863].

The RPD MAY represent interfaces other than those defined above as part of the entPhysicalTable.

11.4.5.4 Entity-MIB Requirements for entPhysicalContainsTable

The purpose of the entPhysicalContainsTable in the RPD and CCAP is to represent the association of multiple downstream and upstream interfaces within the physical construction of the RPD or CCAP. These associations are already modeled in the entPhysicalTable (entPhysicalContainedIn and entPhysicalParentRelPos). The entPhysicalContainsTable provide a more direct relationship of those parent-child associations. Additionally, it may provide mechanisms to indicate other associations like restrictions and configurability of downstream and upstream interfaces within a particular MAC Domain as defined below.

For the purpose of identifying downstream and upstream interfaces within an RF port as well as Channel Blocks, the RPD MAY report in the entPhysicalContainsTable the physical component of Physical Class 'module' as the entPhysicalIndex value for each of the downstream or upstream interface Physical Indexes as the values for entPhysicalChildIndex.

If supported, the CCAP MAY apply the following rules to indicate containment models for MAC Domain and downstream/upstream associations:

- The 'backplane' physical component entries in entPhysicalTable have a valid Physical Index for entPhysicalContainedIn (e.g., the 'chassis' or another 'backplane' Physical Class component).
- The 'backplane' physical components are not referenced by other physical components in entPhysicalTable as their entPhysicalContainedIn value.
- Physical 'backplane' components are the parent index in entPhysicalContainsTable for children indexes representing MAC Domain interfaces, downstream/upstream interfaces, and/or physical components 'modules' that represent RF ports or Channel Blocks. When this set of parent-child entries contains 'modules' (e.g., Channel Blocks) instead of individual US/DS interfaces, it indicates that the complete 'module' is configurable within a single MAC Domain, while the existence of individual 'backplane' - downstream/upstream interfaces parent-children entries in entPhysicalContainsTable indicates that individual channels (even within a Channel Block) can be associated with specific MAC Domains.

The CCAP does not need to report in the entPhysicalContainsTable the MAC Domain downstream/upstream channel hierarchy normally represented in the ifStackTable.

11.4.5.5 Entity-MIB Requirements for entAliasMappingTable

The entAliasMappingTable is used in this specification to associate the physical elements modeled in the Entity MIB with the logical components of the RPD management model. Normally the entAliasLogicalIndexOrZero value is '0' as there are no RPD requirements to support multiple logical entities within the RPD. However, vendors may opt to define multiple logical entities, in which case this object value will be non-zero.

11.4.6 Requirements for Entity Sensor MIB [RFC 3433]

The RPD MAY implement the Entity Sensor MIB [RFC 3433].

For ENTITY-MIB [RFC 4113] entPhysicalTable instances with entPhysicalClass of 'sensor', the RPD MAY implement the entPhySensorTable with the same entPhysicalIndex used in the entPhysicalTable and the entPhySensorType of 'celsius'.

11.4.7 Requirements for Host Resources MIB [RFC 2790]

The RPD MAY implement the HOST-RESOURCES-MIB [RFC 2790].

11.4.8 Requirements for Ethernet Interface MIB [RFC 3635]

The RPD MUST implement [RFC 3635] for each of its Ethernet interfaces.

11.4.9 Requirements for Bridge MIB [RFC 4188]

If the RPD implements link-layer forwarding between Ethernet ports, then it MUST implement the Bridge MIB [RFC 4188] to manage the bridging process and represent state information about the RPD Forwarders using link-layer (bridging) semantics.

If STP is enabled for the RPD, then the RPD implements the dot1dStp scalar group [RFC 4188] and optionally the dot1dStpPortTable [RFC 4188] as specified in Annex A.

11.4.10 Requirements for Internet Protocol MIB [RFC 4293]

The RPD requirements for [RFC 4293] are defined in the following sections.

11.4.10.1 The IP Group

The RPD MUST populate the ipv4InterfaceTable with each Ethernet interface with an assigned IPv4 address. The RPD MAY record other interfaces in the ipv4InterfaceTable which have assigned IPv4 addresses.

The RPD MUST populate the ipv6InterfaceTable with each Ethernet interface with an assigned IPv6 address. The RPD MAY record other interfaces in the ipv6InterfaceTable which have assigned IPv6 addresses.

If the RPD has been configured for a default route, the Routing RPD MUST populate the default router in the ipDefaultRouterTable.

The RPD can populate the ipDefaultRouterTable with an IPv4 and/or IPv6 statically configured default router or a default router learned through a dynamic update mechanism such as a routing protocol update or IPv6 router advertisement message.

11.4.10.2 The ICMP Group

The RPD MUST implement the icmpStatsTable.

The RPD MUST implement the icmpMsgStatsTable.

11.4.11 Requirements for User Datagram Protocol (UDP) MIB [RFC 4113]

The RPD SHOULD implement the UDP-MIB [RFC 4113].

11.4.12 Requirements for Transmission Control Protocol (TCP) MIB [RFC 4022]

The RPD SHOULD implement the TCP group in [RFC 4022].

11.4.13 Requirements for Pseudowire MIB

The RPD and CCAP Core MUST implement a read-only row in *pwTable* for each active L2TPv3 session terminated at the device with object values read as follows:

- *pwType* is read as 'other(0)';
- *pwOwner* is read as 'l2tpControlProtocol (4)';
- *pwPsnType* is read as 'l2tp(2)';
- *pwSetupPriority* is read as 0;
- *pwHoldingPriority* is read as 0;
- *pwPeerAddrType* is read as 'ipv4'or 'ipv6' as appropriate;
- *pwPeerAddr* is the ipv4 or ipv6 address of the remote side of the L2TPv3 session;
- *pwAttachedPwIndex* is read as 0;
- *pwIfIndex* is read as 0 because neither RPD nor CCAP Core implement an ifTable row for L2TPv3 sessions;
- *pwId* is read as 0;
- *pwLocalGroupId* is read as 0;

- *pwGroupAttachmentID, pwLocalAttachmentID,* and *pwRemoteAttachmentId* are all read as NULL (an OCTET STRING of length 0);
- *pwCwPreference* is read as 'false(2)';
- *pwLocalIfMtu* is the device's near-side MTU and is read as the value of the following L2TPV3 AVP transmitted by the device during session setup:

Table 11-11 - PW LOCAL IF MTU

Device	DEPI Downstream PW	UEPI Upstream PW
CCAP Core	DEPI Local MTU AVP (ICRQ)	UEPI Local MTU AVP (ICRQ)
RPD	DEPI Remote MTU AVP (ICRP)	UEPI Remote MTU AVP (ICRP)

- *pwLocalIfString* is read as 'false(2)';
- *pwLocalCapabAdvert* shall have a '1' bit set for bit positions pwStatusIndication(0) and VCCV(1);
- *pwRemoteGroupId* is read as 0;
- *pwCwStatus* is read as cwNotPresent(6);
- *pwRemoteIfMtu* is the far-side MTU of the device's peer and is read as the value of the following L2TPv3 AVP as received by the device during session setup:

Table 11-12 - PW REMOTE IF MTU

Device	DEPI Downstream PW	UEPI Upstream PW
CCAP Core	DEPI Remote MTU AVP (ICRP)	UEPI Remote MTU AVP (ICRP)
RPD	DEPI Local MTU AVP (ICRQ)	UEPI Local MTU AVP (ICRQ)

- *pwRemoteIfString* is read as a 0-length OCTET STRING;
- *pwRemoteCapabilities* is a BITS object with a '1' bit set for only bit positions 'pwStatusIndication(0)' and 'VCCV(1)';
- *pwFragmentCfgSize* is read as 0;
- *pwRmtFragCapability* is read as 'noFrag(0)';
- *pwFcsRetentionCfg* is read as 'fcsRetentionDisable(1)';
- *pwFcsRetentionStatus* is a BITS object with only bit position 'fcsRetentionDisabled(3)' set;
- *pwOutboundLabel* is the L2TPv3 session ID for outgoing data transmitted by the device, and is always the Local Session ID AVP as received by the far side when setting up the session:

Table 11-13 - PW OUT-BOUND LABEL

Device	DEPI Downstream PW	UEPI Upstream PW
CCAP Core	DEPI Local Session ID (ICRP)	UEPI Local Session ID (ICRP)
RPD	DEPI Local Session ID (ICRQ)	UEPI Local Session ID (ICRQ)

- *pwInboundLabel* is the L2TPv3 session ID for incoming data received by the device, and is always the Local Session ID as transmitted to the far side when setting up the session:

Table 11-14 - PW IN-BOUND LABEL

Device	DEPI Downstream PW	UEPI Upstream PW
CCAP Core	DEPI Local Session ID (ICRQ)	UEPI Local Session ID (ICRQ)
RPD	DEPI Local Session ID (ICRP)	UEPI Local Session ID (ICRP)

- *pwName* is the ASCII of the Pseudowire Type Mnemonic for the PseudoWire Type AVP signaled for the session, e.g., "PSPPW".

- *pwDescr* is an ASCII string constructed with the form:
  RemoteEndId=(*pp:mmm*:*ccc*),… { repeated for multiple endpoints }
  where
  pp is the 0-based port number signaled in the RemoteEndId AVP
  m is the channel-type enum value from the RemoteEndId AVP
  ccc is the channel number from the RemoteEndId AVP;
- *pwCreateTime* is the value of sysUpTime when the L2TPv3 session was established;
- *pwUpTime* is the time since the last change of *pwOperStatus* to 'up(1)';
- *pwLastChange* is the value of sysUpTime when the session entered its current *pwOperStatus* state.
- *pwAdminStatus* is always read as 'up(1)';
- *pwOperStatus* is as specified in PW-STD-MIB;
- *pwLocalStatus* is as specified in PW-STD-MIB;
- *pwRemoteStatusCapable* is read as 'notApplicable(1)';
- *pwRemoteStatus* is a BITS object with no bits set;
- *pwTimeElapsed* is read as 0;
- *pwValidIntervals* is read as 0;
- *pwRowStatus* is always read as 'active(1)';
- *pwStorageType* is read as 'volatile(2)';
- *pwOamEnable* is read as 'false(2)'
- *pwGenAGIType, pwGenLocalAIIType,* and *pwGenRemoteAIIType* are all read as 0;

11.4.14 Requirements for DOCSIS Remote PHY MIB [DOCS-RPHY-MIB]

The CCAP Core SHOULD populate the docsRphyDepiSessionInfoTable and docsRphyDepiSessionStatsTable with each DEPI, UEPI, OOB, NDF, and NDR pseudowire it has established with RPDs.

The RPD SHOULD populate the docsRphyDepiSessionInfoTable and docsRphyDepiSessionStatsTable with each DEPI, UEPI, OOB, NDF, and NDR pseudowire it has established with CCAP Cores.

The CCAP Core SHOULD implement the DEPI Latency Measurement (DLM) objects. The RPD MAY implement the DLM objects.

Additional details are expected in a future version of the specification.

11.4.15 Requirements for 8021X-PAE MIB [RFC 4022]

The RPD MUST implement the 8021X-PAE-MIB (additional details are expected in a future version of the specification).

12 OSSI FOR RPD PHYSICAL SECURITY

At the time of this draft release, the entirety of this Section is a placeholder for future updates.

Annex A Detailed MIB Requirements (Normative)

This Annex defines the SNMP MIB modules and MIB variables required for DOCSIS 3.1 CMTS and CCAP devices. Refer to Section 2.1 for the associated MIB files.

Table A-1 - MIB Implementation Support

Requirement Type	Table Notation	Description
Deprecated	D	Deprecated objects are optional. If a vendor chooses to implement the object, the object is expected to be implemented correctly according to the MIB definition. If a vendor chooses not to implement the object, an agent is expected to respond with the appropriate error/exception condition (e.g., 'noSuchObject' for SNMPv2c).
Mandatory	M	The object is expected to be implemented correctly according to the MIB definition.
Not Applicable	NA	Not applicable to the device.
Not Supported	N-Sup	An agent is expected to respond with the appropriate error/exception condition (e.g., 'noSuchObject' for SNMPv2c).
Optional	O	A vendor can choose to implement or not implement the object. If a vendor chooses to implement the object, the object is expected to be implemented correctly according to the MIB definition. If a vendor chooses not to implement the object, an agent is expected to respond with the appropriate error/exception condition (e.g., 'noSuchObject' for SNMPv2c).
Obsolete	Ob	In SNMP convention, obsolete objects should not be implemented. This specification allows vendors to implement or not implement obsolete objects. If a vendor chooses to implement an obsoleted object, the object is expected to be implemented correctly according to the MIB definition. If a vendor chooses not to implement the obsoleted object, the SNMP agent is expected to respond with the appropriate error/exception condition (e.g., 'noSuchObject' for SNMPv2c).

Table A-2 - SNMP Access Requirements

SNMP Access Type	Table Notation	Description
N-Acc	Not Accessible	The object is not accessible and is usually an index in a table
Read Create	RC	The access of the object MUST be implemented as Read-Create
Read Write	RW	The access of the object MUST be implemented as Read-Write
Read Only	RO	The access of the object MUST be implemented as Read-Only
Read Create or Read Only	RC/RO	The access of the object MUST be implemented as either Read-Create or Read-Only as described in the MIB definition
Read Write / Read Only	RW/RO	The access of the object MUST be implemented as either Read-Write or Read-Only as described in the MIB definition
Accessible for SNMP Notifications	Acc-FN	These objects are used for SNMP Notifications by the CMTS and CM SNMP Agents

A.1 MIB Object Details

The CMTS and CCAP instantiates SNMP MIB objects based on its configuration and operational parameters.

The CMTS and CCAP upstream channel types can be categorized as "TDMA/ATDMA upstream" and "SCDMA upstream" and "OFDMA upstream".

Table A-3 - MIB Object Details

DOCS-IF-MIB [RFC 4546]				
Object	**CCAP**	**Access**	**RPD**	**Access**
docsIfDownstreamChannelTable	M	N-Acc	NA	
docsIfDownstreamChannelEntry	M	N-Acc	NA	
docsIfDownChannelId	M	RO	NA	

docsIfDownChannelFrequency	M	RW/RO	NA	
docsIfDownChannelWidth	M	RO	NA	
docsIfDownChannelModulation	M	RW	NA	
docsIfDownChannelInterleave	M	RW	NA	
docsIfDownChannelPower	M	RW/RO	NA	
docsIfDownChannelAnnex	M	RO	NA	
docsIfDownChannelStorageType	M	RO	NA	
docsIfUpstreamChannelTable	M	N-Acc	NA	
docsIfUpstreamChannelEntry	M	N-Acc	NA	
docsIfUpChannelId	M	RO	NA	
docsIfUpChannelFrequency	M	RC	NA	
docsIfUpChannelWidth	M	RC	NA	
docsIfUpChannelModulationProfile	M	RC	NA	
docsIfUpChannelSlotSize	M	RC/RO	NA	
docsIfUpChannelTxTimingOffset	M	RO	NA	
docsIfUpChannelRangingBackoffStart	M	RC	NA	
docsIfUpChannelRangingBackoffEnd	M	RC	NA	
docsIfUpChannelTxBackoffStart	M	RC	NA	
docsIfUpChannelTxBackoffEnd	M	RC	NA	
docsIfUpChannelScdmaActiveCodes	M	RC	NA	
docsIfUpChannelScdmaCodesPerSlot	M	RC	NA	
docsIfUpChannelScdmaFrameSize	M	RC	NA	
docsIfUpChannelScdmaHoppingSeed	M	RC	NA	
docsIfUpChannelType	M	RC	NA	
docsIfUpChannelCloneFrom	M	RC	NA	
docsIfUpChannelUpdate	M	RC	NA	
docsIfUpChannelStatus	M	RC	NA	
Object	**CCAP**	**Access**	**RPD**	**Access**
docsIfUpChannelPreEqEnable	M	RC	NA	
docsIfQosProfileTable	O	N-Acc	NA	
docsIfQosProfileEntry	O	N-Acc	NA	
docsIfQosProfIndex	O	N-Acc	NA	
docsIfQosProfPriority	O	RC/RO	NA	
docsIfQosProfMaxUpBandwidth	O	RC/RO	NA	
docsIfQosProfGuarUpBandwidth	O	RC/RO	NA	
docsIfQosProfMaxDownBandwidth	O	RC/RO	NA	
docsIfQosProfMaxTxBurst	D	RC/RO	NA	
docsIfQosProfBaselinePrivacy	O	RC/RO	NA	

docsIfQosProfStatus	O	RC/RO	NA	
docsIfQosProfMaxTransmitBurst	O	RC/RO	NA	
docsIfQosProfStorageType	O	RO	NA	
docsIfSignalQualityTable	M	N-Acc	NA	
docsIfSignalQualityEntry	M	N-Acc	NA	
docsIfSigQIncludesContention	M	RO	NA	
docsIfSigQUnerroreds	M	RO	NA	
docsIfSigQCorrecteds	M	RO	NA	
docsIfSigQUncorrectables	M	RO	NA	
docsIfSigQSignalNoise	D	RO	NA	
docsIfSigQMicroreflections	M	RO	NA	
docsIfSigQEqualizationData	M	RO	NA	
docsIfSigQExtUnerroreds	M	RO	NA	
docsIfSigQExtCorrecteds	M	RO	NA	
docsIfSigQExtUncorrectables	M	RO	NA	
docsIfDocsisBaseCapability	M	RO	NA	
docsIfCmtsMacTable	M	N-Acc	NA	
docsIfCmtsMacEntry	M	N-Acc	NA	
docsIfCmtsCapabilities	M	RO	NA	
docsIfCmtsSyncInterval	M	RW	NA	
docsIfCmtsUcdInterval	M	RW/RO	NA	
docsIfCmtsMaxServiceIds	M	RO	NA	
docsIfCmtsInsertionInterval	Ob	RW/RO	NA	
docsIfCmtsInvitedRangingAttempts	M	RW/RO	NA	
docsIfCmtsInsertInterval	M	RW/RO	NA	
docsIfCmtsMacStorageType	M	RW/RO	NA	
Object	**CCAP**	**Access**	**RPD**	**Access**
docsIfCmtsStatusTable	D	N-Acc	NA	
docsIfCmtsStatusEntry	D	N-Acc	NA	
docsIfCmtsStatusInvalidRangeReqs	D	RO	NA	
docsIfCmtsStatusRangingAborteds	D	RO	NA	
docsIfCmtsStatusInvalidRegReqs	D	RO	NA	
docsIfCmtsStatusFailedRegReqs	D	RO	NA	
docsIfCmtsStatusInvalidDataReqs	D	RO	NA	
docsIfCmtsStatusT5Timeouts	D	RO	NA	
docsIfCmtsCmStatusTable	D	N-Acc	NA	
docsIfCmtsCmStatusEntry	D	N-Acc	NA	
docsIfCmtsCmStatusIndex	D	N-Acc	NA	

docsIfCmtsCmStatusMacAddress	D	RO	NA	
docsIfCmtsCmStatusIpAddress	D	RO	NA	
docsIfCmtsCmStatusDownChannelIfIndex	D	RO	NA	
docsIfCmtsCmStatusUpChannelIfIndex	D	RO	NA	
docsIfCmtsCmStatusRxPower	D	RO	NA	
docsIfCmtsCmStatusTimingOffset	D	RO	NA	
docsIfCmtsCmStatusEqualizationData	D	RO	NA	
docsIfCmtsCmStatusValue	D	RO	NA	
docsIfCmtsCmStatusUnerroreds	D	RO	NA	
docsIfCmtsCmStatusCorrecteds	D	RO	NA	
docsIfCmtsCmStatusUncorrectables	D	RO	NA	
docsIfCmtsCmStatusSignalNoise	D	RO	NA	
docsIfCmtsCmStatusMicroreflections	D	RO	NA	
docsIfCmtsCmStatusExtUnerroreds	D	RO	NA	
docsIfCmtsCmStatusExtCorrecteds	D	RO	NA	
docsIfCmtsCmStatusExtUncorrectables	D	RO	NA	
docsIfCmtsCmStatusDocsisRegMode	D	RO	NA	
docsIfCmtsCmStatusModulationType	D	RO	NA	
docsIfCmtsCmStatusInetAddressType	D	RO	NA	
docsIfCmtsCmStatusInetAddress	D	RO	NA	
docsIfCmtsCmStatusValueLastUpdate	D	RO	NA	
docsIfCmtsCmStatusHighResolutionTimingOffset	D	RO	NA	
docsIfCmtsServiceTable	M/O	N-Acc	NA	
docsIfCmtsServiceEntry	M/O	N-Acc	NA	
docsIfCmtsServiceId	M/O	N-Acc	NA	
Object	**CCAP**	**Access**	**RPD**	**Access**
docsIfCmtsServiceCmStatusIndex	D	RO	NA	
docsIfCmtsServiceAdminStatus	D	RW/RO	NA	
docsIfCmtsServiceQosProfile	M/O	RO	NA	
docsIfCmtsServiceCreateTime	D	RO	NA	
docsIfCmtsServiceInOctets	D	RO	NA	
docsIfCmtsServiceInPackets	D	RO	NA	
docsIfCmtsServiceNewCmStatusIndex	D	RO	NA	
docsIfCmtsModulationTable	M	N-Acc	NA	
docsIfCmtsModulationEntry	M	N-Acc	NA	
docsIfCmtsModIndex	M	N-Acc	NA	
docsIfCmtsModIntervalUsageCode	M	N-Acc	NA	
docsIfCmtsModControl	M	RC	NA	

docsIfCmtsModType	M	RC	NA	
docsIfCmtsModPreambleLen	M	RC	NA	
docsIfCmtsModDifferentialEncoding	M	RC	NA	
docsIfCmtsModFECErrorCorrection	M	RC	NA	
docsIfCmtsModFECCodewordLength	M	RC	NA	
docsIfCmtsModScramblerSeed	M	RC	NA	
docsIfCmtsModMaxBurstSize	M	RC	NA	
docsIfCmtsModGuardTimeSize	M	RO	NA	
docsIfCmtsModLastCodewordShortened	M	RC	NA	
docsIfCmtsModScrambler	M	RC	NA	
docsIfCmtsModByteInterleaverDepth	M	RC	NA	
docsIfCmtsModByteInterleaverBlockSize	M	RC	NA	
docsIfCmtsModPreambleType	M	RC	NA	
docsIfCmtsModTcmErrorCorrectionOn	M	RC	NA	
docsIfCmtsModScdmaInterleaverStepSize	M	RC	NA	
docsIfCmtsModScdmaSpreaderEnable	M	RO	NA	
docsIfCmtsModScdmaSubframeCodes	M	RC	NA	
docsIfCmtsModChannelType	M	RC	NA	
docsIfCmtsModStorageType	M	RC	NA	
docsIfCmtsQosProfilePermissions	M	RW /RO	NA	
docsIfCmtsMacToCmTable	M	N-Acc	NA	
docsIfCmtsMacToCmEntry	M	N-Acc	NA	
docsIfCmtsCmMac	M	N-Acc	NA	
docsIfCmtsCmPtr	M	RO	NA	
Object	**CCAP**	**Access**	**RPD**	**Access**
docsIfCmtsChannelUtilizationInterval	M	RW	NA	
DocsIfCmtsChannelUtilizationTable	M	N-Acc	NA	
DocsIfCmtsChannelUtilizationEntry	M	N-Acc	NA	
docsIfCmtsChannelUtIfType	M	N-Acc	NA	
docsIfCmtsChannelUtId	M	N-Acc	NA	
docsIfCmtsChannelUtUtilization	M	RO	NA	
docsIfCmtsDownChannelCounterTable	M	N-Acc	NA	
docsIfCmtsDownChannelCounterEntry	M	N-Acc	NA	
docsIfCmtsDownChnlCtrId	M	RO	NA	
docsIfCmtsDownChnlCtrTotalBytes	M	RO	NA	
docsIfCmtsDownChnlCtrUsedBytes	M	RO	NA	
docsIfCmtsDownChnlCtrExtTotalBytes	M	RO	NA	
docsIfCmtsDownChnlCtrExtUsedBytes	M	RO	NA	

docsIfCmtsUpChannelCounterTable	M	N-Acc	NA	
docsIfCmtsUpChannelCounterEntry	M	N-Acc	NA	
docsIfCmtsUpChnlCtrId	M	RO	NA	
docsIfCmtsUpChnlCtrTotalMslots	M	RO	NA	
docsIfCmtsUpChnlCtrUcastGrantedMslots	M	RO	NA	
docsIfCmtsUpChnlCtrTotalCntnMslots	M	RO	NA	
docsIfCmtsUpChnlCtrUsedCntnMslots	M	RO	NA	
docsIfCmtsUpChnlCtrExtTotalMslots	M	RO	NA	
docsIfCmtsUpChnlCtrExtUcastGrantedMslots	M	RO	NA	
docsIfCmtsUpChnlCtrExtTotalCntnMslots	M	RO	NA	
docsIfCmtsUpChnlCtrExtUsedCntnMslots	M	RO	NA	
docsIfCmtsUpChnlCtrCollCntnMslots	M	RO	NA	
docsIfCmtsUpChnlCtrTotalCntnReqMslots	M	RO	NA	
docsIfCmtsUpChnlCtrUsedCntnReqMslots	M	RO	NA	
docsIfCmtsUpChnlCtrCollCntnReqMslots	M	RO	NA	
docsIfCmtsUpChnlCtrTotalCntnReqDataMslots	M	RO	NA	
docsIfCmtsUpChnlCtrUsedCntnReqDataMslots	M	RO	NA	
docsIfCmtsUpChnlCtrCollCntnReqDataMslots	M	RO	NA	
docsIfCmtsUpChnlCtrTotalCntnInitMaintMslots	M	RO	NA	
docsIfCmtsUpChnlCtrUsedCntnInitMaintMslots	M	RO	NA	
docsIfCmtsUpChnlCtrCollCntnInitMaintMslots	M	RO	NA	
docsIfCmtsUpChnlCtrExtCollCntnMslots	M	RO	NA	
docsIfCmtsUpChnlCtrExtTotalCntnReqMslots	M	RO	NA	
Object	**CCAP**	**Access**	**RPD**	**Access**
docsIfCmtsUpChnlCtrExtUsedCntnReqMslots	M	RO	NA	
docsIfCmtsUpChnlCtrExtCollCntnReqMslots	M	RO	NA	
docsIfCmtsUpChnlCtrExtTotalCntnReqDataMslots	M	RO	NA	
docsIfCmtsUpChnlCtrExtUsedCntnReqDataMslots	M	RO	NA	
docsIfCmtsUpChnlCtrExtCollCntnReqDataMslots	M	RO	NA	
docsIfCmtsUpChnlCtrExtTotalCntnInitMaintMslots	M	RO	NA	
docsIfCmtsUpChnlCtrExtUsedCntnInitMaintMslots	M	RO	NA	
docsIfCmtsUpChnlCtrExtCollCntnInitMaintMslots	M	RO	NA	
IF-MIB [RFC 2863]				
Object	**CCAP**	**Access**	**RPD**	**Access**
ifNumber	M	RO	M	RO
ifTableLastChange	M	RO	M	RO
ifTable **Note: The ifTable Counter32 objects are not reflected here; refer to Table 11-9 for details on these objects.**	M	N-Acc	M	N-Acc

ifEntry	M	N-Acc	M	N-Acc
ifIndex	M	RO	M	RO
ifDescr	M	RO	M	RO
ifType	M	RO	M	RO
ifMtu	M	RO	M	RO
ifSpeed	M	RO	M	RO
ifPhysAddress	M	RO	M	RO
ifAdminStatus	M	RW	M	RO
ifOperStatus	M	RO	M	RO
ifLastChange	M	RO	M	RO
ifOutQLen	D	RO	D	RO
ifSpecific	D	RO	D	RO
ifXTable **Note: The ifXTable Counter32 and Counter64 objects are not reflected here; refer to Table 11-9 for details on these objects.**	M	N-Acc	M	N-Acc
ifXEntry	M	N-Acc	M	N-Acc
ifName	M	RO	M	RO
ifLinkUpDownTrapEnable	M	RW	M	RO
ifHighSpeed	M	RO	M	RO
ifPromiscuousMode	M	RW/RO	M	RO
ifConnectorPresent	M	RO	M	RO
ifAlias	M	RW/RO	M	RO
ifCounterDiscontinuityTime	M	RO	M	RO
ifStackTable	M	N-Acc	M	N-Acc
ifStackEntry	M	N-Acc	M	N-Acc
ifStackHigherLayer	M	N-Acc	M	N-Acc
ifStackLowerLayer	M	N-Acc	M	N-Acc
ifStackStatus	M	RC/RO	M	RO
Object	**CCAP**	**Access**	**RPD**	**Access**
ifStackLastChange	M	RC/RO	M	RO
ifRcvAddressTable	O	N-Acc	O	N-Acc
ifRcvAddressEntry	O	N-Acc	O	N-Acc
ifRcvAddressAddress	O	N-Acc	O	N-Acc
ifRcvAddressStatus	O	RC	O	RO
IfRcvAddressType	O	RC	O	RO
Notification				
linkUp	M	Acc-FN	M	Acc-FN
linkDown	M	Acc-FN	M	Acc-FN
ifTestTable	D	N-Acc	D	N-Acc

ifTestEntry	D	N-Acc	D	N-Acc
ifTestId	D	RW	D	RO
ifTestStatus	D	RW	D	RO
ifTestType	D	RW	D	RO
ifTestResult	D	RO	D	RO
ifTestCode	D	RO	D	RO
ifTestOwner	D	RW	D	RO
BRIDGE-MIB [RFC 4188] **Note: Implementation of BRIDGE-MIB is required ONLY if device is a bridging device.** **This MIB needs to be revisited.**				
DOCS-CABLE-DEVICE-MIB [RFC 2669]				
Object	**CCAP**	**Access**	**RPD**	**Access**
docsDevBase				
docsDevRole	O	RO	NA	
docsDevDateTime	M	RW	NA	
docsDevResetNow	O	RW	NA	
docsDevSerialNumber	O	RO	NA	
docsDevSTPControl	O	RW/RO	NA	
docsDevNmAccessTable	O	N-Acc	NA	
docsDevNmAccessEntry	O	N-Acc	NA	
docsDevNmAccessIndex	O	N-Acc	NA	
docsDevNmAccessIp	O	RC	NA	
docsDevNmAccessIpMask	O	RC	NA	
docsDevNmAccessCommunity	O	RC	NA	
docsDevNmAccessControl	O	RC	NA	
docsDevNmAccessInterfaces	O	RC	NA	
docsDevNmAccessStatus	O	RC	NA	
docsDevNmAccessTrapVersion	O	RC	NA	
docsDevSoftware				
docsDevSwServer	D	RW	NA	RO
docsDevSwFilename	O	RW	NA	RO
docsDevSwAdminStatus	O	RW	NA	RO
docsDevSwOperStatus	O	RO	NA	RO
docsDevSwCurrentVers	O	RO	NA	RO
docsDevSwServerAddressType	O	RO	NA	RO
docsDevSwServerAddress	O	RO	NA	RO
docsDevSwServerTransportProtocol	O	RO	NA	RO
docsDevEvent				
docsDevEvControl	M	RW	M	RO

docsDevEvSyslog	D	RW	D	RO
docsDevEvThrottleAdminStatus	M	RW	M	RO
docsDevEvThrottleInhibited	D	RO	D	RO
docsDevEvThrottleThreshold	M	RW	M	RO
docsDevEvThrottleInterval	M	RW	M	RO
docsDevEvControlTable	M	N-Acc	M	N-Acc
docsDevEvControlEntry	M	N-Acc	M	N-Acc
docsDevEvPriority	M	N-Acc	M	N-Acc
docsDevEvReporting	M	RW	M	RO
docsDevEventTable	M	N-Acc	M	N-Acc
docsDevEventEntry	M	N-Acc	M	N-Acc
docsDevEvIndex	M	N-Acc	M	N-Acc
docsDevEvFirstTime	M	RO	M	RO
docsDevEvLastTime	M	RO	M	RO
docsDevEvCounts	M	RO	M	RO
docsDevEvLevel	M	RO	M	RO
docsDevEvId	M	RO	M	RO
docsDevEvText	M	RO	M	RO
docsDevEvSyslogAddressType	M	RW	M	RO
docsDevEvSyslogAddress	M	RW	M	RO
docsDevEvThrottleThresholdExceeded	M	RO	M	RO
docsDevFilter				
docsDevFilterLLCUnmatchedAction	O	RW	NA	
docsDevFilterLLCTable	O	N-Acc	NA	
docsDevFilterLLCEntry	O	N-Acc	NA	
docsDevFilterLLCIndex	O	N-Acc	NA	
docsDevFilterLLCStatus	O	RC	NA	
docsDevFilterLLCIfIndex	O	RC	NA	
docsDevFilterLLCProtocolType	O	RC	NA	
docsDevFilterLLCProtocol	O	RC	NA	
docsDevFilterLLCMatches	O	RO	NA	
Object	**CCAP**	**Access**	**RPD**	**Access**
docsDevFilterIpDefault	O	RW	NA	
docsDevFilterIpTable	D	N-Acc	NA	
docsDevFilterIpEntry	D	N-Acc	NA	
docsDevFilterIpIndex	D	N-Acc	NA	
docsDevFilterIpStatus	D	RC	NA	
docsDevFilterIpControl	D	RC	NA	
docsDevFilterIpIfIndex	D	RC	NA	

docsDevFilterIpDirection	D	RC	NA	
docsDevFilterIpBroadcast	D	RC	NA	
docsDevFilterIpSaddr	D	RC	NA	
docsDevFilterIpSmask	D	RC	NA	
docsDevFilterIpDaddr	D	RC	NA	
docsDevFilterIpDmask	D	RC	NA	
docsDevFilterIpProtocol	D	RC	NA	
docsDevFilterIpSourcePortLow	D	RC	NA	
docsDevFilterIpSourcePortHigh	D	RC	NA	
docsDevFilterIpDestPortLow	D	RC	NA	
docsDevFilterIpDestPortHigh	D	RC	NA	
docsDevFilterIpMatches	D	RO	NA	
docsDevFilterIpTos	D	RC	NA	
docsDevFilterIpTosMask	D	RC	NA	
docsDevFilterIpContinue	D	RC	NA	
docsDevFilterIpPolicyId	D	RC	NA	
docsDevFilterPolicyTable	D	N-Acc	NA	
docsDevFilterPolicyEntry	D	N-Acc	NA	
docsDevFilterPolicyIndex	D	N-Acc	NA	
docsDevFilterPolicyId	D	RC	NA	
docsDevFilterPolicyStatus	D	RC	NA	
docsDevFilterPolicyPtr	D	RC	NA	
docsDevFilterTosTable	D	N-Acc	NA	
docsDevFilterTosEntry	D	N-Acc	NA	
docsDevFilterTosIndex	D	N-Acc	NA	
docsDevFilterTosStatus	D	RC	NA	
docsDevFilterTosAndMask	D	RC	NA	
docsDevFilterTosOrMask	D	RC	NA	
IP-MIB [RFC 4293]				
Object	**CCAP**	**Access**	**RPD**	**Access**
ipv4GeneralGroup				
ipForwarding	M	RW	M	RO
ipDefaultTTL	M	RW	M	RO
ipReasmTimeout	M	RW	M	RO
ipv6GeneralGroup2				
ipv6IpForwarding	M	RW	M	RO
ipv6IpDefaultHopLimit	M	RW	M	RO
ipv4InterfaceTableLastChange	M	RO	M	RO

ipv4InterfaceTable	M	N-Acc	M	N-Acc
ipv4InterfaceEntry	M	N-Acc	M	N-Acc
ipv4InterfaceIfIndex	M	N-Acc	M	N-Acc
ipv4InterfaceReasmMaxSize	M	RO	M	RO
ipv4InterfaceEnableStatus	M	RW	M	RO
ipv4InterfaceRetransmitTime	M	RO	M	RO
ipv6InterfaceTableLastChange	M	RO	M	RO
ipv6InterfaceTable	M	N-Acc	M	N-Acc
ipv6InterfaceEntry	M	N-Acc	M	N-Acc
ipv6InterfaceIfIndex	M	N-Acc	M	N-Acc
ipv6InterfaceReasmMaxSize	M	RO	M	RO
ipv6InterfaceIdentifier	M	RO	M	RO
ipv6InterfaceEnableStatus	M	RW	M	RO
ipv6InterfaceReachableTime	M	RO	M	RO
ipv6InterfaceRetransmitTime	M	RO	M	RO
ipv6InterfaceForwarding	M	RW	M	RO
ipSystemStatsTable	O	N-Acc	O	N-Acc
ipSystemStatsEntry	O	N-Acc	O	N-Acc
ipSystemStatsIPVersion	O	N-Acc	O	N-Acc
ipSystemStatsInReceives	O	RO	O	RO
ipSystemStatsHCInReceives	O	RO	O	RO
ipSystemStatsInOctets	O	RO	O	RO
ipSystemStatsHCInOctets	O	RO	O	RO
ipSystemStatsInHdrErrors	O	RO	O	RO
ipSystemStatsInNoRoutes	O	RO	O	RO
ipSystemStatsInAddrErrors	O	RO	O	RO
ipSystemStatsInUnknownProtos	O	RO	O	RO
ipSystemStatsInTruncatedPkts	O	RO	O	RO
ipSystemStatsInForwDatagrams	O	RO	O	RO
ipSystemStatsHCInForwDatagrams	O	RO	O	RO
ipSystemStatsReasmReqds	O	RO	O	RO
ipSystemStatsReasmOKs	O	RO	O	RO
ipSystemStatsReasmFails	O	RO	O	RO
ipSystemStatsInDiscards	O	RO	O	RO
ipSystemStatsInDelivers	O	RO	O	RO
ipSystemStatsHCInDelivers	O	RO	O	RO
ipSystemStatsOutRequests	O	RO	O	RO
ipSystemStatsHCOutRequests	O	RO	O	RO
ipSystemStatsOutNoRoutes	O	RO	O	RO

ipSystemStatsOutForwDatagrams	O	RO	O	RO
ipSystemStatsHCOutForwDatagrams	O	RO	O	RO
ipSystemStatsOutDiscards	O	RO	O	RO
ipSystemStatsOutFragReqds	O	RO	O	RO
ipSystemStatsOutFragOKs	O	RO	O	RO
ipSystemStatsOutFragFails	O	RO	O	RO
ipSystemStatsOutFragCreates	O	RO	O	RO
ipSystemStatsOutTransmits	O	RO	O	RO
ipSystemStatsHCOutTransmits	O	RO	O	RO
ipSystemStatsOutOctets	O	RO	O	RO
ipSystemStatsHCOutOctets	O	RO	O	RO
ipSystemStatsInMcastPkts	O	RO	O	RO
ipSystemStatsHCInMcastPkts	O	RO	O	RO
ipSystemStatsInMcastOctets	O	RO	O	RO
ipSystemStatsHCInMcastOctets	O	RO	O	RO
ipSystemStatsOutMcastPkts	O	RO	O	RO
ipSystemStatsHCOutMcastPkts	O	RO	O	RO
ipSystemStatsOutMcastOctets	O	RO	O	RO
ipSystemStatsHCOutMcastOctets	O	RO	O	RO
ipSystemStatsInBcastPkts	O	RO	O	RO
ipSystemStatsHCInBcastPkts	O	RO	O	RO
ipSystemStatsOutBcastPkts	O	RO	O	RO
ipSystemStatsHCOutBcastPkts	O	RO	O	RO
ipSystemStatsDiscontinuityTime	O	RO	O	RO
ipSystemStatsRefreshRate	O	RO	O	RO
Object	**CCAP**	**Access**	**RPD**	**Access**
ipIfStatsTableLastChange	O	RO	O	RO
ipIfStatsTable **Note: This table is required ONLY if routing is implemented.**	M	N-Acc	NA	
ipIfStatsEntry	M	N-Acc	NA	
ipIfStatsIPVersion	M	N-Acc	NA	
ipIfStatsIfIndex	M	N-Acc	NA	
ipIfStatsInReceives	M	RO	NA	
ipIfStatsHCInReceives	M	RO	NA	
ipIfStatsInOctets	M	RO	NA	
ipIfStatsHCInOctets	M	RO	NA	
ipIfStatsInHdrErrors	M	RO	NA	
ipIfStatsInNoRoutes	M	RO	NA	

ipIfStatsInAddrErrors	M	RO	NA	
ipIfStatsInUnknownProtos	M	RO	NA	
ipIfStatsInTruncatedPkts	M	RO	NA	
ipIfStatsInForwDatagrams	M	RO	NA	
ipIfStatsHCInForwDatagrams	M	RO	NA	
ipIfStatsReasmReqds	M	RO	NA	
ipIfStatsReasmOKs	M	RO	NA	
ipIfStatsReasmFails	M	RO	NA	
ipIfStatsInDiscards	M	RO	NA	
ipIfStatsInDelivers	M	RO	NA	
ipIfStatsHCInDelivers	M	RO	NA	
ipIfStatsOutRequests	M	RO	NA	
ipIfStatsHCOutRequests	M	RO	NA	
ipIfStatsOutForwDatagrams	M	RO	NA	
ipIfStatsHCOutForwDatagrams	M	RO	NA	
ipIfStatsOutDiscards	M	RO	NA	
ipIfStatsOutFragReqds	M	RO	NA	
ipIfStatsOutFragOKs	M	RO	NA	
ipIfStatsOutFragFails	M	RO	NA	
ipIfStatsOutFragCreates	M	RO	NA	
ipIfStatsOutTransmits	M	RO	NA	
ipIfStatsHCOutTransmits	M	RO	NA	
ipIfStatsOutOctets	M	RO	NA	
ipIfStatsHCOutOctets	M	RO	NA	
ipIfStatsInMcastPkts	M	RO	NA	
ipIfStatsHCInMcastPkts	M	RO	NA	
ipIfStatsInMcastOctets	M	RO	NA	
ipIfStatsHCInMcastOctets	M	RO	NA	
ipIfStatsOutMcastPkts	M	RO	NA	
ipIfStatsHCOutMcastPkts	M	RO	NA	
ipIfStatsOutMcastOctets	M	RO	NA	
ipIfStatsHCOutMcastOctets	M	RO	NA	
ipIfStatsInBcastPkts	M	RO	NA	
ipIfStatsHCInBcastPkts	M	RO	NA	
ipIfStatsOutBcastPkts	M	RO	NA	
ipIfStatsHCOutBcastPkts	M	RO	NA	
ipIfStatsDiscontinuityTime	M	RO	NA	
ipIfStatsRefreshRate	M	RO	NA	

ipAddressPrefixTable **Note: This table is required ONLY if routing is implemented.**	M	N-Acc	NA	
ipAddressPrefixEntry	M	N-Acc	NA	
ipAddressPrefixIfIndex	M	N-Acc	NA	
ipAddressPrefixType	M	N-Acc	NA	
ipAddressPrefixPrefix	M	N-Acc	NA	
ipAddressPrefixLength	M	N-Acc	NA	
ipAddressPrefixOrigin	M	RO	NA	
ipAddressPrefixOnLinkFlag	M	RO	NA	
ipAddressPrefixAutonomousFlag	M	RO	NA	
ipAddressPrefixAdvPreferredLifetime	M	RO	NA	
ipAddressPrefixAdvValidLifetime	M	RO	NA	
ipAddressSpinLock	M	RW	NA	
ipAddressTable	M	N-Acc	M	N-Acc
ipAddressEntry	M	N-Acc	M	N-Acc
ipAddressAddrType	M	N-Acc	M	N-Acc
ipAddressAddr	M	N-Acc	M	N-Acc
ipAddressIfIndex	M	RO	M	RO
ipAddressType	M	RO	M	RO
ipAddressPrefix	M	RO	M	RO
ipAddressOrigin	M	RO	M	RO
ipAddressStatus	M	RO	M	RO
ipAddressCreated	M	RO	M	RO
ipAddressLastChanged	M	RO	M	RO
ipAddressRowStatus	M	RO	M	RO
ipAddressStorageType	M	RO	M	RO
ipNetToPhysicalTable **Note: This table is required ONLY if routing is implemented.**	M	N-Acc	M	
ipNetToPhysicalEntry	M	N-Acc	M	
ipNetToPhysicalIfIndex	M	N-Acc	M	
ipNetToPhysicalNetAddressType	M	N-Acc	M	
ipNetToPhysicalNetAddress	M	N-Acc	M	
ipNetToPhysicalPhysAddress	M	RC	M	
ipNetToPhysicalLastUpdated	M	RO	M	
ipNetToPhysicalType	M	RC	M	
ipNetToPhysicalState	M	RO	M	
ipNetToPhysicalRowStatus	M	RC	M	

ipDefaultRouterTable **Note: This table is required ONLY if routing is implemented.**	M	N-Acc	M	
ipDefaultRouterEntry	M	N-Acc	M	
ipDefaultRouterAddressType	M	N-Acc	M	
ipDefaultRouterAddress	M	N-Acc	M	
ipDefaultRouterIfIndex	M	N-Acc	M	
ipDefaultRouterLifetime	M	RC	M	
ipDefaultRouterPreference	M	RO	M	
ipv6RouterAdvertGroup			NA	
ipv6RouterAdvertSpinLock	O	RW	NA	
ipv6RouterAdvertTable **Note: This table is required ONLY if routing is implemented.**	M	N-Acc	NA	
ipv6RouterAdvertEntry	M	N-Acc	NA	
ipv6RouterAdvertIfIndex	M	N-Acc	NA	
ipv6RouterAdvertSendAdverts	M	RC	NA	
ipv6RouterAdvertMaxInterval	M	RC	NA	
ipv6RouterAdvertMinInterval	M	RC	NA	
ipv6RouterAdvertManagedFlag	M	RC	NA	
ipv6RouterAdvertOtherConfigFlag	M	RC	NA	
ipv6RouterAdvertLinkMTU	M	RC	NA	
ipv6RouterAdvertReachableTime	M	RC	NA	
ipv6RouterAdvertRetransmitTime	M	RC	NA	
ipv6RouterAdvertCurHopLimit	M	RC	NA	
ipv6RouterAdvertDefaultLifetime	M	RC	NA	
ipv6RouterAdvertRowStatus	M	RC	NA	
icmpStatsTable	M	N-Acc	M	N-Acc
icmpStatsEntry	M	N-Acc	M	N-Acc
icmpStatsIPVersion	M	N-Acc	M	N-Acc
icmpStatsInMsgs	M	RO	M	RO
icmpStatsInErrors	M	RO	M	RO
icmpStatsOutMsgs	M	RO	M	RO
icmpStatsOutErrors	M	RO	M	RO
icmpMsgStatsTable	M	N-Acc	M	N-Acc
icmpMsgStatsEntry	M	N-Acc	M	N-Acc
icmpMsgStatsIPVersion	M	N-Acc	M	N-Acc
icmpMsgStatsType	M	N-Acc	M	N-Acc
icmpMsgStatsInPkts	M	RO	M	RO
icmpMsgStatsOutPkts	M	RO	M	RO

UDP-MIB [RFC 4113]				
Object	**CCAP**	**Access**	**RPD**	**Access**
UDPGroup				
udpInDatagrams	O	RO	O	RO
udpNoPorts	O	RO	O	RO
udpInErrors	O	RO	O	RO
udpOutDatagrams	O	RO	O	RO
udpEndpointTable	O	N-Acc	O	N-Acc
udpEndpointEntry	O	N-Acc	O	N-Acc
udpEndpointLocalAddressType	O	N-Acc	O	N-Acc
udpEndpointLocalAddress	O	N-Acc	O	N-Acc
udpEndpointLocalPort	O	N-Acc	O	N-Acc
udpEndpointRemoteAddressType	O	N-Acc	O	N-Acc
udpEndpointRemoteAddress	O	N-Acc	O	N-Acc
udpEndpointRemotePort	O	N-Acc	O	N-Acc
udpEndpointInstance	O	N-Acc	O	N-Acc
udpEndpointProcess	O	RO	O	RO
TCP-MIB [RFC 4022]				
Object	**CCAP**	**Access**	**RPD**	**Access**
tcpBaseGroup				
tcpRtoAlgorithm	O	RO	O	RO
tcpRtoMin	O	RO	O	RO
tcpRtoMax	O	RO	O	RO
tcpMaxConn	O	RO	O	RO
tcpActiveOpens	O	RO	O	RO
tcpPassiveOpens	O	RO	O	RO
tcpAttemptFails	O	RO	O	RO
tcpEstabResets	O	RO	O	RO
tcpCurrEstab	O	RO	O	RO
tcpInSegs	O	RO	O	RO
tcpOutSegs	O	RO	O	RO
tcpRetransSegs	O	RO	O	RO
tcpInErrs	O	RO	O	RO
tcpOutRsts	O	RO	O	RO
tcpHCGroup				
tcpHCInSegs	O	RO	O	RO
tcpHCOutSegs	O	RO	O	RO
tcpConnectionTable	O	N-Acc	O	N-Acc

tcpConnectionEntry	O	N-Acc	O	N-Acc
tcpConnectionLocalAddressType	O	N-Acc	O	N-Acc
tcpConnectionLocalAddress	O	N-Acc	O	N-Acc
tcpConnectionLocalPort	O	N-Acc	O	N-Acc
tcpConnectionRemAddressType	O	N-Acc	O	N-Acc
tcpConnectionRemAddress	O	N-Acc	O	N-Acc
tcpConnectionRemPort	O	N-Acc	O	N-Acc
tcpConnectionState	O	RW	O	RO
tcpConnectionProcess	O	RO	O	RO
tcpListenerTable	O	N-Acc	O	N-Acc
tcpListenerEntry	O	N-Acc	O	N-Acc
tcpListenerLocalAddressType	O	N-Acc	O	N-Acc
tcpListenerLocalAddress	O	N-Acc	O	N-Acc
tcpListenerLocalPort	O	N-Acc	O	N-Acc
tcpListenerProcess	O	RO	O	RO
SNMPv2-MIB [RFC 3418]				
Object	**CCAP**	**Access**	**RPD**	**Access**
SystemGroup				
sysDescr	M	RO	M	RO
sysObjectID	M	RO	M	RO
sysUpTime	M	RO	M	RO
sysContact	M	RW	M	RO
sysName	M	RW	M	RO
sysLocation	M	RW	M	RO
sysServices	M	RO	M	RO
sysORLastChange	M	RO	M	RO
sysORTable	M	N-Acc	M	N-Acc
sysOREntry	M	N-Acc	M	N-Acc
sysORIndex	M	N-Acc	M	N-Acc
sysORID	M	RO	M	RO
sysORDescr	M	RO	M	RO
sysORUpTime	M	RO	M	RO
SNMPGroup				
snmpInPkts	M	RO	M	RO
snmpInBadVersions	M	RO	M	RO
snmpOutPkts	Ob	RO	Ob	RO
snmpInBadCommunityNames	M	RO	M	RO
snmpInBadCommunityUses	M	RO	M	RO

snmpInASNParseErrs	M	RO	M	RO
snmpInTooBigs	Ob	RO	Ob	RO
snmpInNoSuchNames	Ob	RO	Ob	RO
snmpInBadValues	Ob	RO	Ob	RO
snmpInReadOnlys	Ob	RO	Ob	RO
snmpInGenErrs	Ob	RO	Ob	RO
snmpInTotalReqVars	Ob	RO	Ob	RO
snmpInTotalSetVars	Ob	RO	Ob	RO
snmpInGetRequests	Ob	RO	Ob	RO
snmpInGetNexts	Ob	RO	Ob	RO
snmpInSetRequests	Ob	RO	Ob	RO
snmpInGetResponses	Ob	RO	Ob	RO
snmpInTraps	Ob	RO	Ob	RO
snmpOutTooBigs	Ob	RO	Ob	RO
snmpOutNoSuchNames	Ob	RO	Ob	RO
snmpOutBadValues	Ob	RO	Ob	RO
snmpOutGenErrs	Ob	RO	Ob	RO
snmpOutGetRequests	Ob	RO	Ob	RO
snmpOutGetNexts	Ob	RO	Ob	RO
snmpOutSetRequests	Ob	RO	Ob	RO
snmpOutGetResponses	Ob	RO	Ob	RO
snmpOutTraps	Ob	RO	Ob	RO
snmpEnableAuthenTraps	M	RW	M	RO
snmpSilentDrops	M	RO	M	RO
snmpProxyDrops	M	RO	M	RO
snmpTrapsGroup				
coldStart	M	Acc-FN	M	Acc-FN
warmStart	O	Acc-FN	O	Acc-FN
authenticationFailure	M	Acc-FN	M	Acc-FN
snmpSetGroup				
snmpSetSerialNo	M	RW	M	RO
Etherlike-MIB [RFC 3635]				
Object	**CCAP**	**Access**	**RPD**	**Access**
dot3StatsTable	M	N-Acc	M	N-Acc
dot3StatsEntry	M	N-Acc	M	N-Acc
dot3StatsIndex	M	RO	M	RO
dot3StatsAlignmentErrors	M	RO	M	RO
dot3StatsFCSErrors	M	RO	M	RO

dot3StatsInternalMacTransmitErrors	M	RO	M	RO
dot3StatsFrameTooLongs	M	RO	M	RO
dot3StatsInternalMacReceiveErrors	M	RO	M	RO
dot3StatsSymbolErrors	M	RO	M	RO
dot3StatsSingleCollisionFrames	O	RO	O	RO
dot3StatsMultipleCollisionFrames	O	RO	O	RO
dot3StatsDeferredTransmissions	O	RO	O	RO
dot3StatsLateCollisions	O	RO	O	RO
dot3StatsExcessiveCollisions	O	RO	O	RO
dot3StatsCarrierSenseErrors	O	RO	O	RO
dot3StatsDuplexStatus	O	RO	O	RO
dot3StatsSQETestErrors	N-Sup		N-Sup	
dot3CollTable	O	N-Acc	O	N-Acc
dot3CollEntry	O	N-Acc	O	N-Acc
dot3CollCount	O	NA	O	NA
dot3CollFrequencies	O	RO	O	RO
dot3ControlTable	O	N-Acc	O	N-Acc
dot3ControlEntry	O	N-Acc	O	N-Acc
dot3ControlFunctionsSupported	O	RO	O	RO
dot3ControlInUnknownOpcodes	O	RO	O	RO
dot3PauseTable	O	N-Acc	O	N-Acc
dot3PauseEntry	O	N-Acc	O	N-Acc
dot3PauseAdminMode	O	RW	O	RO
dot3PauseOperMode	O	RO	O	RO
dot3InPauseFrames	O	RO	O	RO
dot3OutPauseFrames	O	RO	O	RO
DOCS-IFEXT2-MIB [DOCS-IFEXT2-MIB]				
Object	**CCAP**	**Access**	**RPD**	**Access**
docsIfExt2CmtsObjects				
docsIfExt2CmtsMscGlobalEnable	M	RW	NA	
docsIfExt2CmtsCmMscStatusTable	O	N-Acc	NA	
docsIfExt2CmtsCmMscStatusEntry	O	N-Acc	NA	
docsIfExt2CmtsCmMscStatusPowerShortfall	O	RO	NA	
docsIfExt2CmtsCmMscStatusCodeRatio	O	RO	NA	
docsIfExt2CmtsCmMscStatusMaximumScheduledCodes	O	RO	NA	
docsIfExt2CmtsCmMscStatusPowerHeadroom	O	RO	NA	
docsIfExt2CmtsCmMscStatusMeasuredSNR	O	RO	NA	
docsIfExt2CmtsCmMscStatusEffectiveSNR	O	RO	NA	

docsIfExt2CmtsUpChannelMscTable	O	N-Acc	NA	
docsIfExt2CmtsUpChannelMscEntry	O	N-Acc	NA	
docsIfExt2CmtsUpChannelMscState	O	RW	NA	
docsIfExt2CmtsUpChannelMSCTotalCMs	O	RO	NA	
docsIfExt2CmtsUpChannelMSCLimitIUC1	O	RO	NA	
docsIfExt2CmtsUpChannelMSCMinimumValue	O	RW	NA	
docsIfExt2CmtsUpChannelTable	O	N-Acc	NA	
docsIfExt2CmtsUpChannelEntry	O	N-Acc	NA	
docsIfExt2CmtsUpChannelTotalCMs	O	RO	NA	
HOST-RESOURCES-MIB [RFC 2790]				
Object	**CCAP**	**Access**	**RPD**	**Access**
hrDeviceTable	O	N-Acc	O	N-Acc
hrDeviceEntry	O	N-Acc	O	N-Acc
hrDeviceIndex	O	RO	O	RO
hrDeviceType	O	RO	O	RO
hrDeviceDescr	O	RO	O	RO
hrDeviceID	O	RO	O	RO
hrDeviceStatus	O	RO	O	RO
hrDeviceErrors	O	RO	O	RO
hrSystem				
hrMemorySize	O	RO	O	RO
hrStorageTable	O	N-Acc	O	N-Acc
hrStorageEntry	O	N-Acc	O	N-Acc
hrStorageIndex	O	RO	O	RO
hrStorageType	O	RO	O	RO
hrStorageDescr	O	RO	O	RO
hrStorageAllocationUnits	O	RO	O	RO
hrStorageSize	O	RO	O	RO
hrStorageUsed	O	RO	O	RO
hrStorageAllocationFailures	O	RO	O	RO
hrSWRunTable	O	N-Acc	O	N-Acc
hrSWRunEntry	O	N-Acc	O	N-Acc
hrSWRunIndex	O	RO	O	RO
hrSWRunName	O	RO	O	RO
hrSWRunID	O	RO	O	RO
hrSWRunPath	O	RO	O	RO
hrSWRunParameters	O	RO	O	RO
hrSWRunType	O	RO	O	RO

hrSWRunStatus	O	RO	O	RO
hrSWRunPerfTable	O	N-Acc	O	N-Acc
hrSWRunPerfEntry	O	N-Acc	O	N-Acc
hrSWRunIndex	O	N-Acc	O	N-Acc
hrSWRunPerfCPU	O	RO	O	RO
hrSWRunPerfMem	O	RO	O	RO
hrProcessorTable	O	N-Acc	O	N-Acc
hrProcessorEntry	O	N-Acc	O	N-Acc
hrProcessorFrwID	O	RO	O	RO
hrProcessorLoad	O	RO	O	RO
ENTITY-MIB [RFC 4133]				
Object	**CCAP**	**Access**	**RPD**	**Access**
entPhysicalTable	O	N-Acc	M	N-Acc
entPhysicalEntry	O	N-Acc	M	N-Acc
entPhysicalIndex	O	N-Acc	M	N-Acc
entPhysicalDescr	O	RO	M	RO
entPhysicalVendorType	O	RO	M	RO
entPhysicalContainedIn	O	RO	M	RO
entPhysicalClass	O	RO	M	RO
entPhysicalParentRelPos	O	RO	M	RO
entPhysicalName	O	RO	M	RO
entPhysicalHardwareRev	O	RO	O	RO
entPhysicalFirmwareRev	O	RO	O	RO
entPhysicalSoftwareRev	O	RO	O	RO
entPhysicalSerialNum	O	RO/RW	O	RO
entPhysicalMfgName	O	RO	O	RO
entPhysicalModelName	O	RO	O	RO
entPhysicalAlias	O	RO/RW	O	RO
entPhysicalAssetID	O	RO/RW	O	RO
entPhysicalIsFRU	O	RO	O	RO
entPhysicalMfgDate	O	RO	O	RO
entPhysicalUris	O	RW	O	RO
entLogicalTable	O	N-Acc	O	N-Acc
entLogicalEntry	O	N-Acc	O	N-Acc
entLogicalIndex	O	N-Acc	O	N-Acc
entLogicalDescr	O	RO	O	RO
entLogicalType	O	RO	O	RO
entLogicalCommunity	D	RO	D	RO

entLogicalTAddress	O	RO	O	RO
entLogicalTDomain	O	RO	O	RO
entLogicalContextEngineID	O	RO	O	RO
entLogicalContextName	O	RO	O	RO
entLPMappingTable	O	N-Acc	O	N-Acc
entLPMappingEntry	O	N-Acc	O	N-Acc
entLPPhysicalIndex	O	RO	O	RO
entAliasMappingTable	O	N-Acc	O	N-Acc
entAliasMappingEntry	O	N-Acc	O	N-Acc
entAliasLogicalIndexOrZero	O	N-Acc	O	N-Acc
entAliasMappingIdentifier	O	RO	O	RO
entPhysicalContainsTable	O	N-Acc	O	N-Acc
entPhysicalContainsEntry	O	N-Acc	O	N-Acc
entPhysicalChildIndex	O	RO	O	RO
General Group				
entLastChangeTime	O	RO	O	RO
Notification				
entConfigChange	O	Acc-FN	O	Acc-FN
ENTITY-SENSOR-MIB [RFC 3433]				
Object	**CCAP**	**Access**	**RPD**	**Access**
entPhySensorTable	O	N-Acc	M	N-Acc
entPhySensorEntry	O	N-Acc	M	N-Acc
entPhySensorType	O	RO	M	RO
entPhySensorScale	O	RO	M	RO
entPhySensorPrecision	O	RO	M	RO
entPhySensorValue	O	RO	M	RO
entPhySensorOperStatus	O	RO	M	RO
entPhySensorUnitsDisplay	O	RO	M	RO
entPhySensorValueTimeStamp	O	RO	M	RO
entPhySensorValueUpdateRate	O	RO	M	RO
SNMP-USM-DH-OBJECTS-MIB [RFC 2786]				
Object	**CCAP**	**Access**	**RPD**	**Access**
usmDHParameters	O	RW	O	RO
usmDHUserKeyTable	O	N-Acc	O	N-Acc
usmDHUserKeyEntry	O	N-Acc	O	N-Acc
usmDHUserAuthKeyChange	O	RC	O	RO
usmDHUserOwnAuthKeyChange	O	RC	O	RO
usmDHUserPrivKeyChange	O	RC	O	RO

usmDHUserOwnPrivKeyChange	O	RC	O	RO
usmDHKickstartTable	O	N-Acc	O	N-Acc
usmDHKickstartEntry	O	N-Acc	O	N-Acc
usmDHKickstartIndex	O	N-Acc	O	N-Acc
usmDHKickstartMyPublic	O	RO	O	RO
usmDHKickstartMgrPublic	O	RO	O	RO
usmDHKickstartSecurityName	O	RO	O	RO
SNMP-VIEW-BASED-ACM-MIB [RFC 2575]				
Object	**CCAP**	**Access**	**RPD**	**Access**
vacmContextTable	O	N-Acc	O	N-Acc
vacmContextEntry	O	N-Acc	O	N-Acc
vacmContextName	O	RO	O	RO
vacmSecurityToGroupTable	O	N-Acc	O	N-Acc
vacmSecurityToGroupEntry	O	N-Acc	O	N-Acc
vacmSecurityModel	O	N-Acc	O	N-Acc
vacmSecurityName	O	N-Acc	O	N-Acc
vacmGroupName	O	RC	O	RO
vacmSecurityToGroupStorageType	O	RC	O	RO
vacmSecurityToGroupStatus	O	RC	O	RO
vacmAccessTable	O	N-Acc	O	N-Acc
vacmAccessEntry	O	N-Acc	O	N-Acc
vacmAccessContextPrefix	O	N-Acc	O	N-Acc
vacmAccessSecurityModel	O	N-Acc	O	N-Acc
vacmAccessSecurityLevel	O	N-Acc	O	N-Acc
vacmAccessContextMatch	O	RC	O	RO
vacmAccessReadViewName	O	RC	O	RO
vacmAccessWriteViewName	O	RC	O	RO
vacmAccessNotifyViewName	O	RC	O	RO
vacmAccessStorageType	O	RC	O	RO
vacmAccessStatus	O	RC	O	RO
vacmViewSpinLock	O	RW	O	RO
vacmViewTreeFamilyTable	O	N-Acc	O	N-Acc
vacmViewTreeFamilyEntry	O	N-Acc	O	N-Acc
vacmViewTreeFamilyViewName	O	N-Acc	O	N-Acc
vacmViewTreeFamilySubtree	O	N-Acc	O	N-Acc
vacmViewTreeFamilyMask	O	RC	O	RO
vacmViewTreeFamilyType	O	RC	O	RO
vacmViewTreeFamilyStorageType	O	RC	O	RO

vacmViewTreeFamilyStatus	O	RC	O	RO
SNMP-COMMUNITY-MIB [RFC 3584]				
Object	**CCAP**	**Access**	**RPD**	**Access**
snmpCommunityTable	M	N-Acc	M	N-Acc
snmpCommunityEntry	M	N-Acc	M	N-Acc
snmpCommunityIndex	M	N-Acc	M	N-Acc
snmpCommunityName	M	RC	M	RO
snmpCommunitySecurityName	M	RC	M	RO
snmpCommunityContextEngineID	M	RC	M	RO
snmpCommunityContextName	M	RC	M	RO
snmpCommunityTransportTag	M	RC	M	RO
snmpCommunityStorageType	M	RC	M	RO
snmpCommunityStatus	M	RC	M	RO
snmpTargetAddrExtTable	M	N-Acc	M	N-Acc
snmpTargetAddrExtEntry	M	N-Acc	M	N-Acc
snmpTargetAddrTMask	M	RC	M	RO
snmpTargetAddrMMS	M	RC	M	RO
snmpTrapAddress	O	ACC-FN	O	ACC-FN
snmpTrapCommunity	O	ACC-FN	O	ACC-FN
SNMP-FRAMEWORK-MIB [RFC 3411]				
Object	**CCAP**	**Access**	**RPD**	**Access**
snmpEngineGroup				
snmpEngineID	M	RO	M	RO
snmpEngineBoots	M	RO	M	RO
snmpEngineTime	M	RO	M	RO
snmpEngineMaxMessageSize	M	RO	M	RO
SNMP-MPD-MIB [RFC 3412]				
Object	**CCAP**	**Access**	**RPD**	**Access**
snmpMPDStats				
snmpUnknownSecurityModels	M	RO	M	RO
snmpInvalidMsgs	M	RO	M	RO
snmpUnknownPDUHandlers	M	RO	M	RO
SNMP Applications [RFC 2573]				
Object	**CCAP**	**Access**	**RPD**	**Access**
snmpTargetSpinLock	M	RW	M	RO
snmpTargetAddrTable	M	N-Acc	M	N-Acc
snmpTargetAddrEntry	M	N-Acc	M	N-Acc
snmpTargetAddrName	M	N-Acc	M	N-Acc

snmpTargetAddrTDomain	M	RC	M	RO
snmpTargetAddrTAddress	M	RC	M	RO
snmpTargetAddrTimeout	M	RC	M	RO
snmpTargetAddrRetryCount	M	RC	M	RO
snmpTargetAddrTagList	M	RC	M	RO
snmpTargetAddrParams	M	RC	M	RO
snmpTargetAddrStorageType	M	RC	M	RO
snmpTargetAddrRowStatus	M	RC	M	RO
snmpTargetParamsTable	M	N-Acc	M	N-Acc
snmpTargetParamsEntry	M	N-Acc	M	N-Acc
snmpTargetParamsName	M	N-Acc	M	N-Acc
snmpTargetParamsMPModel	M	RC	M	RO
snmpTargetParamsSecurityModel	M	RC	M	RO
snmpTargetParamsSecurityName	M	RC	M	RO
snmpTargetParamsSecurityLevel	M	RC	M	RO
snmpTargetParamsStorageType	M	RC	M	RO
snmpTargetParamsRowStatus	M	RC	M	RO
snmpUnavailableContexts	M	RO	M	RO
snmpUnknownContexts	M	RO	M	RO
snmpNotifyTable	M	N-Acc	M	N-Acc
snmpNotifyEntry	M	N-Acc	M	N-Acc
snmpNotifyName	M	N-Acc	M	N-Acc
snmpNotifyTag	M	RC	M	RO
snmpNotifyType	M	RC	M	RO
snmpNotifyStorageType	M	RC	M	RO
snmpNotifyRowStatus	M	RC	M	RO
snmpNotifyFilterProfileTable	M	N-Acc	M	N-Acc
snmpNotifyFilterProfileEntry	M	N-Acc	M	N-Acc
snmpNotifyFilterProfileName	M	RC	M	RO
snmpNotifyFilterProfileStorType	M	RC	M	RO
snmpNotifyFilterProfileRowStatus	M	RC	M	RO
snmpNotifyFilterTable	M	N-Acc	M	N-Acc
snmpNotifyFilterEntry	M	N-Acc	M	N-Acc
snmpNotifyFilterSubtree	M	N-Acc	M	N-Acc
snmpNotifyFilterMask	M	RC	M	RO
snmpNotifyFilterType	M	RC	M	RO
snmpNotifyFilterStorageType	M	RC	M	RO
snmpNotifyFilterRowStatus	M	RC	M	RO
SNMP-USER-BASED-SM-MIB [RFC 3414]				

Object	CCAP	Access	RPD	Access
usmStats				
usmStatsUnsupportedSecLevels	O	RO	O	RO
usmStatsNotInTimeWindows	O	RO	O	RO
usmStatsUnknownUserNames	O	RO	O	RO
usmStatsUnknownEngineIDs	O	RO	O	RO
usmStatsWrongDigests	O	RO	O	RO
usmStatsDecryptionErrors	O	RO	O	RO
usmUser				
usmUserSpinLock	O	RW	O	RO
usmUserTable	O	N-Acc	O	N-Acc
usmUserEntry	O	N-Acc	O	N-Acc
usmUserEngineID	O	N-Acc	O	N-Acc
usmUserName	O	N-Acc	O	N-Acc
usmUserSecurityName	O	RO	O	RO
usmUserCloneFrom	O	RC	O	RO
usmUserAuthProtocol	O	RC	O	RO
usmUserAuthKeyChange	O	RC	O	RO
usmUserOwnAuthKeyChange	O	RC	O	RO
usmUserPrivProtocol	O	RC	O	RO
usmUserPrivKeyChange	O	RC	O	RO
usmUserOwnPrivKeyChange	O	RC	O	RO
usmUserPublic	O	RC	O	RO
usmUserStorageType	O	RC	O	RO
usmUserStatus	O	RC	O	RO
DOCS-IF3-MIB [DOCS-IF3-MIB]				
Object	**CCAP**	**Access**	**RPD**	**Access**
docsIf3MdNodeStatusTable	M	N-Acc	NA	
docsIf3MdNodeStatusEntry	M	N-Acc	NA	
docsIf3MdNodeStatusNodeName	M	N-Acc	NA	
docsIf3MdNodeStatusMdCmSgId	M	N-Acc	NA	
docsIf3MdNodeStatusMdDsSgId	M	RO	NA	
docsIf3MdNodeStatusMdUsSgId	M	RO	NA	
docsIf3MdDsSgStatusTable	M	N-Acc	NA	
docsIf3MdDsSgStatusEntry	M	N-Acc	NA	
docsIf3MdDsSgStatusMdDsSgId	M	N-Acc	NA	
docsIf3MdDsSgStatusChSetId	M	RO	NA	
docsIf3MdUsSgStatusTable	M	N-Acc	NA	

docsIf3MdUsSgStatusEntry	M	N-Acc	NA	
docsIf3MdUsSgStatusMdUsSgId	M	N-Acc	NA	
docsIf3MdUsSgStatusChSetId	M	RO	NA	
docsIf3CmtsCmRegStatusTable	M	N-Acc	NA	
docsIf3CmtsCmRegStatusEntry	M	N-Acc	NA	
docsIf3CmtsCmRegStatusId	M	N-Acc	NA	
docsIf3CmtsCmRegStatusMacAddr	M	RO	NA	
docsIf3CmtsCmRegStatusIPv6Addr	M	RO	NA	
docsIf3CmtsCmRegStatusIPv6LinkLocal	M	RO	NA	
docsIf3CmtsCmRegStatusIPv4Addr	M	RO	NA	
docsIf3CmtsCmRegStatusValue	M	RO	NA	
docsIf3CmtsCmRegStatusMdIfIndex	M	RO	NA	
docsIf3CmtsCmRegStatusMdCmSgId	M	RO	NA	
docsIf3CmtsCmRegStatusRcpId	M	RO	NA	
docsIf3CmtsCmRegStatusRccStatusId	M	RO	NA	
docsIf3CmtsCmRegStatusRcsId	M	RO	NA	
docsIf3CmtsCmRegStatusTcsId	M	RO	NA	
docsIf3CmtsCmRegStatusQosVersion	M	RO	NA	
docsIf3CmtsCmRegStatusLastRegTime	M	RO	NA	
docsIf3CmtsCmRegStatusAddrResolutionReqs	M	RO	NA	
docsIf3CmtsCmRegStatusEnergyMgtEnabled	M	RO	NA	
docsIf3CmtsCmRegStatusEnergyMgtOperStatus	M	RO	NA	
docsIf3CmtsCmUsStatusTable	M	N-Acc	NA	
docsIf3CmtsCmUsStatusEntry	M	N-Acc	NA	
docsIf3CmtsCmUsStatusChIfIndex	M	N-Acc	NA	
docsIf3CmtsCmUsStatusModulationType	M	RO	NA	
docsIf3CmtsCmUsStatusRxPower	M	RO	NA	
docsIf3CmtsCmUsStatusSignalNoise	M	RO	NA	
docsIf3CmtsCmUsStatusMicroreflections	M	RO	NA	
docsIf3CmtsCmUsStatusEqData	M	RO	NA	
docsIf3CmtsCmUsStatusUnerroreds	M	RO	NA	
docsIf3CmtsCmUsStatusCorrecteds	M	RO	NA	
docsIf3CmtsCmUsStatusUncorrectables	M	RO	NA	
docsIf3CmtsCmUsStatusHighResolutionTimingOffset	M	RO	NA	
docsIf3CmtsCmUsStatusIsMuted	M	RO	NA	
docsIf3CmtsCmUsStatusRangingStatus	M	RO	NA	
docsIf3MdCfgTable	M	N-Acc	NA	
docsIf3MdCfgEntry	M	N-Acc	NA	

docsIf3MdCfgMddInterval	M	RW	NA	
docsIf3MdCfgIpProvMode	M	RW	NA	
docsIf3MdCfgCmStatusEvCtlEnabled	M	RW	NA	
docsIf3MdCfgUsFreqRange	M	RW	NA	
docsIf3MdCfgMcastDsidFwdEnabled	O	RW	NA	
docsIf3MdCfgMultRxChModeEnabled	M	RW	NA	
docsIf3MdCfgMultTxChModeEnabled	M	RW	NA	
docsIf3MdCfgEarlyAuthEncrCtrl	M	RW	NA	
docsIf3MdCfgTftpProxyEnabled	M	RW	NA	
docsIf3MdCfgSrcAddrVerifEnabled	M	RW	NA	
docsIf3MdCfgDownChannelAnnex	M	RW	NA	
docsIf3MdCfgCmUdcEnabled	M	RW	NA	
docsIf3MdCfgSendUdcRulesEnabled	O	RW	NA	
docsIf3MdCfgServiceTypeIdList	M	RW	NA	
docsIf3MdCfgBpi2EnforceCtrl	M	RW	NA	
docsIf3MdCfgEnergyMgt1x1Enabled	M	RW	NA	
docsIf3MdChCfgTable	M	N-Acc	NA	
docsIf3MdChCfgEntry	M	N-Acc	NA	
docsIf3MdChCfgChIfIndex	M	N-Acc	NA	
docsIf3MdChCfgIsPriCapableDs	M	RC	NA	
docsIf3MdChCfgChId	M	RC	NA	
docsIf3MdChCfgSfProvAttrMask	M	RC	NA	
docsIf3MdChCfgRowStatus	M	RC	NA	
docsIf3MdUsToDsChMappingTable	M	N-Acc	NA	
docsIf3MdUsToDsChMappingEntry	M	N-Acc	NA	
docsIf3MdUsToDsChMappingUsIfIndex	M	N-Acc	NA	
docsIf3MdUsToDsChMappingDsIfIndex	M	N-Acc	NA	
docsIf3MdUsToDsChMappingMdIfIndex	M	RO	NA	
docsIf3DsChSetTable	M	N-Acc	NA	
docsIf3DsChSetEntry	M	N-Acc	NA	
docsIf3DsChSetId	M	N-Acc	NA	
docsIf3DsChSetChList	M	RO	NA	
docsIf3UsChSetTable	M	N-Acc	NA	
docsIf3UsChSetEntry	M	N-Acc	NA	
docsIf3UsChSetId	M	N-Acc	NA	
docsIf3UsChSetChList	M	RO	NA	
docsIf3BondingGrpCfgTable	M	N-Acc	NA	
docsIf3BondingGrpCfgEntry	M	N-Acc	NA	
docsIf3BondingGrpCfgDir	M	N-Acc	NA	

docsIf3BondingGrpCfgCfgId	M	N-Acc	NA	
docsIf3BondingGrpCfgChList	M	RC	NA	
docsIf3BondingGrpCfgSfProvAttrMask	M	RC	NA	
docsIf3BondingGrpCfgDsidReseqWaitTime	M	RC	NA	
docsIf3BondingGrpCfgDsidReseqWarnThrshld	M	RC	NA	
docsIf3BondingGrpCfgRowStatus	M	RC	NA	
docsIf3DsBondingGrpStatusTable	M	N-Acc	NA	
docsIf3DsBondingGrpStatusEntry	M	N-Acc	NA	
docsIf3DsBondingGrpStatusChSetId	M	N-Acc	NA	
docsIf3DsBondingGrpStatusMdDsSgId	M	RO	NA	
docsIf3DsBondingGrpStatusCfgId	M	RO	NA	
docsIf3UsBondingGrpStatusTable	M	N-Acc	NA	
docsIf3UsBondingGrpStatusEntry	M	N-Acc	NA	
docsIf3UsBondingGrpStatusChSetId	M	N-Acc	NA	
docsIf3UsBondingGrpStatusMdUsSgId	M	RO	NA	
docsIf3UsBondingGrpStatusCfgId	M	RO	NA	
docsIf3RccCfgTable	M	N-Acc	NA	
docsIf3RccCfgEntry	M	N-Acc	NA	
docsIf3RccCfgRcpId	M	N-Acc	NA	
docsIf3RccCfgRccCfgId	M	N-Acc	NA	
docsIf3RccCfgVendorSpecific	M	RC	NA	
docsIf3RccCfgDescription	M	RC	NA	
docsIf3RccCfgRowStatus	M	RC	NA	
docsIf3RxChCfgTable	M	N-Acc	NA	
docsIf3RxChCfgEntry	M	N-Acc	NA	
docsIf3RxChCfgRcId	M	N-Acc	NA	
docsIf3RxChCfgChIfIndex	M	RO	NA	
docsIf3RxChCfgPrimaryDsIndicator	M	RC	NA	
docsIf3RxChCfgRcRmConnectivityId	M	RC	NA	
docsIf3RxChCfgRowStatus	M	RC	NA	
docsIf3RxModuleCfgTable	M	N-Acc	NA	
docsIf3RxModuleCfgEntry	M	N-Acc	NA	
docsIf3RxModuleCfgRmId	M	N-Acc	NA	
docsIf3RxModuleCfgRmRmConnectivityId	M	RC	NA	
docsIf3RxModuleCfgFirstCenterFrequency	M	RC	NA	
docsIf3RxModuleCfgRowStatus	M	RC	NA	
docsIf3RccStatusTable	M	N-Acc	NA	
docsIf3RccStatusEntry	M	N-Acc	NA	
docsIf3RccStatusRcpId	M	N-Acc	NA	

docsIf3RccStatusRccStatusId	M	N-Acc	NA	
docsIf3RccStatusRccCfgId	M	RO	NA	
docsIf3RccStatusValidityCode	M	RO	NA	
docsIf3RccStatusValidityCodeText	M	RO	NA	
docsIf3RxChStatusTable	M	N-Acc	NA	
docsIf3RxChStatusEntry	M	N-Acc	NA	
docsIf3RxChStatusRcId	M	N-Acc	NA	
docsIf3RxChStatusChIfIndex	M	RO	NA	
docsIf3RxChStatusPrimaryDsIndicator	M	RO	NA	
docsIf3RxChStatusRcRmConnectivityId	M	RO	NA	
docsIf3RxModuleStatusTable	M	N-Acc	NA	
docsIf3RxModuleStatusEntry	M	N-Acc	NA	
docsIf3RxModuleStatusRmId	M	N-Acc	NA	
docsIf3RxModuleStatusRmRmConnectivityId	M	RO	NA	
docsIf3RxModuleStatusFirstCenterFrequency	M	RO	NA	
docsIf3SignalQualityExtTable	M	N-Acc	NA	
docsIf3SignalQualityExtEntry	M	N-Acc	NA	
docsIf3SignalQualityExtRxMER	M	RO	NA	
docsIf3SignalQualityExtRxMerSamples	M	RO	NA	
docsIf3CmtsSignalQualityExtTable	M	N-Acc	NA	
docsIf3CmtsSignalQualityExtEntry	M	N-Acc	NA	
docsIf3CmtsSignalQualityExtCNIR	M	RO	NA	
docsIf3CmtsSignalQualityExtExpectedRxSignalPower	M	RW	NA	
docsIf3CmtsSpectrumAnalysisMeasTable	M	N-Acc	NA	
docsIf3CmtsSpectrumAnalysisMeasEntry	M	N-Acc	NA	
docsIf3CmtsSpectrumAnalysisMeasAmplitudeData	M	RO	NA	
docsIf3CmtsSpectrumAnalysisMeasTimeInterval	M	RO	NA	
docsIf3CmtsSpectrumAnalysisMeasRowStatus	M	RC	NA	
docsIf3UsChExtTable	M	N-Acc	NA	
docsIf3UsChExtEntry	M	N-Acc	NA	
docsIf3UsChExtSacCodeHoppingSelectionMode	M	RO	NA	
docsIf3UsChExtScdmaSelectionStringActiveCodes	M	RO	NA	
docsIf3CmtsCmCtrlCmd			NA	
docsIf3CmtsCmCtrlCmdMacAddr	M	RW	NA	
docsIf3CmtsCmCtrlCmdMuteUsChId	M	RW	NA	
docsIf3CmtsCmCtrlCmdMuteInterval	M	RW	NA	
docsIf3CmtsCmCtrlCmdDisableForwarding	M	RW	NA	
docsIf3CmtsCmCtrlCmdCommit	M	RW	NA	

docsIf3CmtsEventCtrlTable	M	N-Acc	NA	
docsIf3CmtsEventCtrlEntry	M	N-Acc	NA	
docsIf3CmtsEventCtrlEventId	M	N-Acc	NA	
docsIf3CmtsEventCtrlStatus	M	RC	NA	
docsIf3CmtsCmEmStatsTable	M	N-Acc	NA	
docsIf3CmtsCmEmStatsEntry	M	N-Acc	NA	
docsIf3CmtsCmEmStatsEm1x1ModeTotalDuration	M	RO	NA	
Notifications			NA	
docsIf3CmtsEventNotif	M	Notif	NA	
CLAB-TOPO-MIB [CLAB-TOPO-MIB]				
Object	**CCAP**	**Access**	**RPD**	**Access**
clabTopoFiberNodeCfgTable	M	N-Acc	NA	
clabTopoFiberNodeCfgEntry	M	N-Acc	NA	
clabTopoFiberNodeCfgNodeName	M	N-Acc	NA	
clabTopoFiberNodeCfgNodeDescr	M	RC	NA	
clabTopoFiberNodeCfgRowStatus	M	RC	NA	
clabTopoChFnCfgTable	M	N-Acc	NA	
clabTopoChFnCfgEntry	M	N-Acc	NA	
clabTopoChFnCfgNodeName	M	N-Acc	NA	
clabTopoChFnCfgChIfIndex	M	N-Acc	NA	
clabTopoChFnCfgRowStatus	M	RC	NA	
PW-STD-MIB [RFC 5601]				
Object	**CCAP**	**Access**	**RPD**	**Access**
pwStdMIB	O	N-Acc	O	N-Acc
pwObjects	O	N-Acc	O	N-Acc
pwIndexNext	O	RO	O	RO
pwTable	M	N-Acc	M	N-Acc
pwEntry	M	N-Acc	M	N-Acc
pwIndex	M	N-Acc	M	N-Acc
pwType	M	RO	M	RO
pwOwner	M	RO	M	RO
pwPsnType	M	RO	M	RO
pwSetUpPriority	M	RO	M	RO
pwHoldingPriority	M	RO	M	RO
pwPeerAddrType	M	RO	M	RO
pwPeerAddr	M	RO	M	RO
pwAttachedPwIndex	M	RO	M	RO
pwIfIndex	M	RO	M	RO

pwID	M	RO	M	RO
pwLocalGroupID	M	RO	M	RO
pwGroupAttachmentID	M	RO	M	RO
pwLocalAttachmentID	M	RO	M	RO
pwRemoteAttachmentID	M	RO	M	RO
pwCwPreference	M	RO	M	RO
pwLocalIfMtu	M	RO	M	RO
pwLocalIfString	M	RO	M	RO
pwLocalCapabAdvert	M	RO	M	RO
pwRemoteGroupID	M	RO	M	RO
pwCwStatus	M	RO	M	RO
pwRemoteIfMtu	M	RO	M	RO
pwRemoteIfString	M	RO	M	RO
pwRemoteCapabilities	O	RO	O	RO
pwFragmentCfgSize	M	RO	M	RO
pwRmtFragCapability	O	RO	O	RO
pwFcsRetentionCfg	M	RO	M	RO
pwFcsRetentionStatus	O	RO	O	RO
pwOutboundLabel	M	RO	M	RO
pwInboundLabel	M	RO	M	RO
pwName	M	RO	M	RO
pwDescr	M	RO	M	RO
pwCreateTime	M	RO	M	RO
pwUpTime	M	RO	M	RO
pwLastChange	M	RO	M	RO
pwAdminStatus	M	RO	M	RO
pwOperStatus	M	RO	M	RO
pwLocalStatus	M	RO	M	RO
pwRemoteStatusCapable	O	RO	O	RO
pwRemoteStatus	O	RO	O	RO
pwTimeElapsed	O	RO	O	RO
pwValidIntervals	O	RO	O	RO
pwRowStatus	M	RO	M	RO
pwStorageType	M	RO	M	RO
pwOamEnable	M	RO	M	RO
pwGenAGIType	M	RO	M	RO
pwGenLocalAIIType	M	RO	M	RO
pwGenRemoteAIIType	M	RO	M	RO
pwPerfCurrentInHCPackets	M	RO	M	RO

pwPerfCurrentInHCBytes	M	RO	M	RO
pwPerfCurrentOutHCPackets	M	RO	M	RO
pwPerfCurrentOutHCBytes	M	RO	M	RO
pwPerfCurrentInPackets	O	RO	O	RO
pwPerfCurrentInBytes	O	RO	O	RO
pwPerfCurrentOutPackets	O	RO	O	RO
pwPerfCurrentOutBytes	O	RO	O	RO
pwPerfIntervalTable	O	N-Acc	O	N-Acc
pwPerfIntervalEntry	O	N-Acc	O	N-Acc
pwPerfIntervalNumber	O	N-Acc	O	N-Acc
pwPerfIntervalValidData	O	RO	O	RO
pwPerfIntervalTimeElapsed	O	RO	O	RO
pwPerfIntervalInHCPackets	O	RO	O	RO
pwPerfIntervalInHCBytes	O	RO	O	RO
pwPerfIntervalOutHCPackets	O	RO	O	RO
pwPerfIntervalOutHCBytes	O	RO	O	RO
pwPerfIntervalInPackets	O	RO	O	RO
pwPerfIntervalInBytes	O	RO	O	RO
pwPerfIntervalOutPackets	O	RO	O	RO
pwPerfIntervalOutBytes	O	RO	O	RO
pwPerf1DayIntervalTable	O	RO	O	RO
pwPerf1DayIntervalEntry	O	RO	O	RO
pwPerf1DayIntervalNumber	O	RO	O	RO
pwPerf1DayIntervalValidData	O	RO	O	RO
pwPerf1DayIntervalTimeElapsed	O	RO	O	RO
pwPerf1DayIntervalInHCPackets	O	RO	O	RO
pwPerf1DayIntervalInHCBytes	O	RO	O	RO
pwPerf1DayIntervalOutHCPackets	O	RO	O	RO
pwPerf1DayIntervalOutHCBytes	O	RO	O	RO
pwPerfTotalErrorPackets	O	RO	O	RO
pwIndexMappingTable	O	N-Acc	O	N-Acc
pwIndexMappingEntry	O	N-Acc	O	N-Acc
pwIndexMappingPwType	O	N-Acc	O	N-Acc
pwIndexMappingPwID	O	N-Acc	O	N-Acc
pwIndexMappingPeerAddrType	O	N-Acc	O	N-Acc
pwIndexMappingPeerAddr	O	N-Acc	O	N-Acc
pwIndexMappingPwIndex	O	RO	O	RO
pwPeerMappingTable	O	N-Acc	O	N-Acc
pwPeerMappingEntry	O	N-Acc	O	N-Acc

pwPeerMappingPeerAddrType	O	N-Acc	O	N-Acc
pwPeerMappingPeerAddr	O	N-Acc	O	N-Acc
pwPeerMappingPwType	O	N-Acc	O	N-Acc
pwPeerMappingPwID	O	N-Acc	O	N-Acc
pwPeerMappingPwIndex	O	RO	O	RO
pwUpDownNotifEnable	O	RO	O	RO
pwDeletedNotifEnable	O	RO	O	RO
pwNotifRate	O	RO	O	RO
pwGenFecIndexMappingTable	O	RO	O	RO
pwGenFecIndexMappingEntry	O	RO	O	RO
pwGenFecIndexMappingAGIType	O	RO	O	RO
pwGenFecIndexMappingAGI	O	RO	O	RO
pwGenFecIndexMappingLocalAIIType	O	RO	O	RO
pwGenFecIndexMappingLocalAII	O	RO	O	RO
pwGenFecIndexMappingRemoteAIIType	O	RO	O	RO
pwGenFecIndexMappingRemoteAII	O	RO	O	RO
pwGenFecIndexMappingPwIndex	O	RO	O	RO
pwDown	O	Notif	O	Notif
pwUp	O	Notif	O	Notif
pwDeleted	O	Notif	O	Notif
BFD-STD-MIB				
Object	**CCAP**	**Access**	**RPD**	**Access**
Additional Details in future version of specification				
IEEE8021X-PAE-MIB				
Object	**CCAP**	**Access**	**RPD**	**Access**
Additional Details in future version of specification				
IEEE8021-SECY-MIB	**Note: Optional but if implemented MUST implement requirements below.**			
Object	**CCAP**	**Access**	**RPD**	**Access**
SecY Management				
secyIfTable	O	N-Acc	O	N-Acc
secyIfEntry	O	N-Acc	O	N-Acc
secyIfInterfaceIndex	O	N-Acc	O	N-Acc
secyIfMaxPeerSCs	M	RO	M	RO
secyIfRxMaxKeys	M	RO	M	RO
secyIfTxMaxKeys	M	RO	M	RO
secyIfProtectFramesEnable	M	RW	M	RW
secyIfValidateFrames	M	RW	M	RW
secyIfReplayProtectEnable	M	RW	M	RW

secyIfReplayProtectWindow	M	RW	M	RW
secyIfCurrentCipherSuite	M	RW	M	RW
secyIfAdminPt2PtMAC	M	RW	M	RW
secyIfOperPt2PtMAC	M	RO	M	RO
secyIfIncludeSCIEnable	M	RW	M	RW
secyIfUseESEnable	M	RW	M	RW
secyIfUseSCBEnable	M	RW	M	RW
Tx SC Management				
secyTxSCTable	O	N-Acc	O	N-Acc
secyTxSCEntry	O	N-Acc	O	N-Acc
secyTxSCI	M	RO	M	RO
secyTxSCState	M	RO	M	RO
secyTxSCEncodingSA	M	RO	M	RO
secyTxSCEncipheringSA	M	RO	M	RO
secyTxSCCreatedTime	M	RO	M	RO
secyTxSCStartedTime	M	RO	M	RO
secyTxSCStoppedTime	M	RO	M	RO
Tx SA Management				
secyTxSATable	O	N-Acc	O	N-Acc
secyTxSAEntry	O	N-Acc	O	N-Acc
secyTxSA	O	N-Acc	O	N-Acc
secyTxSAState	M	RO	M	RO
secyTxSANextPN	M	RO	M	RO
secyTxSAConfidentiality	M	RO	M	RO
secyTxSASAKUnchanged	M	RO	M	RO
secyTxSACreatedTime	M	RO	M	RO
secyTxSAStartedTime	M	RO	M	RO
secyTxSAStoppedTime	M	RO	M	RO
Rx SC Management				
secyRxSCTable	O	N-Acc	O	N-Acc
secyRxSCEntry	O	N-Acc	O	N-Acc
secyRxSCI	O	N-Acc	O	N-Acc
secyRxSCState	M	RO	M	RO
secyRxSCCurrentSA	M	RO	M	RO
secyRxSCCreatedTime	M	RO	M	RO
secyRxSCStartedTime	M	RO	M	RO
secyRxSCStoppedTime	M	RO	M	RO
Rx SA Management				
secyRxSATable	O	N-Acc	O	N-Acc

secyRxSAEntry	O	N-Acc	O	N-Acc
secyRxSA	O	N-Acc	O	N-Acc
secyRxSAState	M	RO	M	RO
secyRxSANextPN	M	RO	M	RO
secyRxSASAKUnchanged	M	RO	M	RO
secyRxSACreatedTime	M	RO	M	RO
secyRxSAStartedTime	M	RO	M	RO
secyRxSAStoppedTime	M	RO	M	RO
SecY Selectable Cipher Suites				
secyCipherSuiteTable	O	N-Acc	O	N-Acc
secyCipherSuiteEntry	O	N-Acc	O	N-Acc
secyCipherSuiteIndex	O	N-Acc	O	N-Acc
secyCipherSuiteId	M	RC	M	RC
secyCipherSuiteName	M	RC	M	RC
secyCipherSuiteCapability	M	RC	M	RC
secyCipherSuiteProtection	M	RC	M	RC
secyCipherSuiteProtectionOffset	M	RC	M	RC
secyCipherSuiteDataLengthChange	M	RC	M	RC
secyCipherSuiteICVLength	M	RC	M	RC
secyCipherSuiteRowStatus	M	RC	M	RC
TX SA Statistics				
secyTxSAStatsTable	O	N-Acc	O	N-Acc
secyTxSAStatsEntry	O	N-Acc	O	N-Acc
secyTxSAStatsProtectedPkts	M	RO	M	RO
secyTxSAStatsEncryptedPkts	M	RO	M	RO
TX SC Statistics				
secyTxSCStatsTable	O	N-Acc	O	N-Acc
secyTxSCStatsEntry	O	N-Acc	O	N-Acc
secyTxSCStatsProtectedPkts	M	RO	M	RO
secyTxSCStatsEncryptedPkts	M	RO	M	RO
secyTxSCStatsOctetsProtected	M	RO	M	RO
secyTxSCStatsOctetsEncrypted	M	RO	M	RO
RX SA Statistics				
secyRxSAStatsTable	O	N-Acc	O	N-Acc
secyRxSAStatsEntry	O	N-Acc	O	N-Acc
secyRxSAStatsUnusedSAPkts	M	RO	M	RO
secyRxSAStatsNoUsingSAPkts	M	RO	M	RO
secyRxSAStatsNotValidPkts	M	RO	M	RO
secyRxSAStatsInvalidPkts	M	RO	M	RO

secyRxSAStatsOKPkts	M	RO	M	RO
RX SC Statistics				
secyRxSCStatsTable	O	N-Acc	O	N-Acc
secyRxSCStatsEntry	O	N-Acc	O	N-Acc
secyRxSCStatsUnusedSAPkts	M	RO	M	RO
secyRxSCStatsNoUsingSAPkts	M	RO	M	RO
secyRxSCStatsLatePkts	M	RO	M	RO
secyRxSCStatsNotValidPkts	M	RO	M	RO
secyRxSCStatsInvalidPkts	M	RO	M	RO
secyRxSCStatsDelayedPkts	M	RO	M	RO
secyRxSCStatsUncheckedPkts	M	RO	M	RO
secyRxSCStatsOKPkts	M	RO	M	RO
secyRxSCStatsOctetsValidated	M	RO	M	RO
secyRxSCStatsOctetsDecrypted	M	RO	M	RO
SECY Statistics				
secyStatsTable	O	N-Acc	O	N-Acc
secyStatsEntry	O	N-Acc	O	N-Acc
secyStatsTxUntaggedPkts	M	RO	M	RO
secyStatsTxTooLongPkts	M	RO	M	RO
secyStatsRxUntaggedPkts	M	RO	M	RO
secyStatsRxNoTagPkts	M	RO	M	RO
secyStatsRxBadTagPkts	M	RO	M	RO
secyStatsRxUnknownSCIPkts	M	RO	M	RO
secyStatsRxNoSCIPkts	M	RO	M	RO
secyStatsRxOverrunPkts	M	RO	M	RO
DOCS-RPHY-MIB [DOCS-RPHY-MIB]				
Object	**CCAP**	**Access**	**RPD**	**Access**
docsRphyDepiSessionInfoTable	O	N-Acc	O	N-Acc
docsRphyDepiSessionInfoEntry	O	N-Acc	O	N-Acc
docsRphyDepiSessionInfoUdpPort	M	RO	M	RO
docsRphyDepiSessionInfoMaxPayload	M	RO	M	RO
docsRphyDepiSessionInfoPathPayload	M	RO	M	RO
docsRphyDepiSessionInfoIncludeDOCSISMsgs	M	RO	M	RO
docsRphyDepiSessionInfoRsrcAlloccResp	M	RO	M	RO
docsRphyDepiSessionInfoConnCtrlID	M	RO	M	RO
docsRphyDepiSessionInfoSessionID	M	RO	M	RO
docsRphyDepiSessionInfoOwner	M	RO	M	RO
docsRphyDepiSessionInfoState	M	RO	M	RO

docsRphyDepiSessionInfoErrorCode	M	RO	M	RO
docsRphyDepiSessionInfoCreationTime	M	RO	M	RO
docsRphyDepiSessionInfoStorage	M	RO	M	RO
docsRphyDepiSesstionStatsTable	O	N-Acc	O	N-Acc
docsRphyDepiSessionStatsEntry	O	N-Acc	O	N-Acc
docsRphyDepiSessionStatsOutOfSeqquencePkts	M	RO	M	RO
docsRphyDepiSessionCinLatency	O	RO	O	RO
docsRphyDepiSessionCinLastValue	O	RO	O	RO
docsRphyDepiSessionCinLastValueIfIndex	O	RO	O	RO
docsRphyDepiSessionCinLatencyValueLastTime	O	RO	O	RO
docsRphyDepiSessionCinLatencyPerfTable	O	N-Acc	O	N-Acc
docsRphyDepiSessionCinLatencyPerfEntry	O	N-Acc	O	N-Acc
docsRphyDepiSessionCinLatencyPerfIntervalSeq	M	RO	M	RO
docsRphyDepiSessionCinLatencyPerfValue	M	RO	M	RO
docsRphyDepiSessionCinLatencyTime	M	RO	M	RO

Annex B Format and Content for Event, SYSLOG, and SNMP Notification (Normative)

Details to be further defined.

Table B-1 in this annex summarizes the format and content for event, syslog, and SNMP notifications required for DOCSIS 3.1-compliant CCAP.

Each row specifies a possible event that may appear in the CMTS and CCAP. These events are to be reported by a cable device through local event logging, and may be accompanied by syslog or SNMP notification.

The "Process" and "Sub-Process" columns indicate in which stage the event happens. The "CMTS/CCAP Priority" column indicates the priority the event is assigned in the CMTS and CCAP. These priorities are the same as is reported in the docsDevEvLevel object in the cable device MIB [RFC 4639] and in the LEVEL field of the syslog.

The "Event Message" column specifies the event text, which is reported in the docsDevEvText object of the cable device MIB and the text field of the syslog. The "Message Notes And Details" column provides additional information about the event text in the "Event Message" column. Some of the text fields include variable information. The variables are explained in the "Message Notes And Details" column. For some events the "Message Notes And Details" column may include the keyword <Deprecated> to indicate this event is being deprecated and its implementation is optional. For events where the "Event Message" or "Message Notes and Details" column includes either <P1> or <P2>, there is a single space between the value as defined by the <P1> or <P2> and the preceding text.

Example SNMP Notification and Syslog message "Event Message" text string for Event ID 69020900:

SNMP CVC Validation Failure SNMP Manager: 10.50.1.11;CM-MAC=00:22:ce:03:f4:da;CMTS-MAC=00:15:20:00:25:ab;CM-QOS=1.1;CM-VER=3.0;

This specification defines the following keywords as part of the "Event Message" column:

"<TAGS>" (without the quotes) corresponds to:

For the CMTS (without the quotes): ";<CM-MAC>;<CM-QOS>;<CM-VER>;<CMTS-VER>;"

Where:

<CM-MAC>: CM MAC Address;

Format*: "CM-MAC=xx:xx:xx:xx:xx:xx"

<CMTS-MAC>: CMTS MAC Address;

Format*: "CMTS-MAC=xx:xx:xx:xx:xx:xx"

<CM-QOS>: CM DOCSIS QOS Version;

Format*: "CM-QOS=1.0" or "CM-QOS=1.1"

<CM-VER>: CM DOCSIS Version;

Format*: "CM-VER=1.1" or "CM-VER=2.0" or "CM-VER=3.0" or "CM-VER=3.1"

<CMTS-VER>: CMTS DOCSIS Version;

Format*: "CMTS-VER=1.1" or "CMTS-VER=2.0" or "CMTS-VER=3.0" or "CMTS-VER=3.1"

(*) without the quotes

The CCAP Core MUST support all events defined in Annex D of [CCAP-OSSIv3.1]. The CCAP Core MUST support all mandatory events as defined in Table B-1.

The RPD MUST support all events defined in Table B-2.

Example SNMP Notification and Syslog message "Event Message" text string for Event ID 69010100:

```
SW Download INIT - Via NMS SW file: junk.bin - SW server: 10.50.1.11;CM-
MAC=00:22:ce:03:f4:da;CMTS-MAC=00:15:20:00:25:ab;CM-QOS=1.1;CM-VER=3.0;
```

The CCAP Core and RPD MAY append additional vendor-specific text to the end of the event text reported in the docsDevEvText object and the syslog text field.

The "Error Code Set" column specifies the error code. The "Event ID" column indicates a unique identification number for the event, which is assigned to the docsDevEvId object in the cable device MIB and the <eventId> field of the syslog. The "Notification Name" column specifies the SNMP notification, which notifies this event to an SNMP notification receiver.

The syslog format, as well as the rules to uniquely generate an event ID from the error code, are described in Section 10.2.2.1.3 of this specification.

Table B-1 - CCAP Core Event Format and Content

Process	Sub-Process	CMTS/CCAP Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
Authentication and Encryption	**1**						
	Authentication		Authentication error to RPD: <P1>: <P2>	P1 = RPD ID P2 = authentication error description			docsDevCmtsEventNotif
	IKE Mutual Authentication						docsDevCmtsEventNotif

Table B-2 - RPD Event Format and Content

Process	Sub-Process	CMTS/CCAP Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
Authentication and Encryption	**1**						**docsDevCmtsEventNotif**
							docsDevCmtsEventNotif
Connectivity							
		Critical	Connection lost - secondary CCAP Core: <P1>	P1 = CCAP Core ID			docsDevCmtsEventNotif
		Critical	Connection lost – primary CCAP Core				docsDevCmtsEventNotif
DHCP and TOD and TFTP	**3**						
DHCP		Error	DHCP RENEW sent – No response for <P1><TAGS>	P1=IPv4 or IPv6			docsDevCmtsEventNotif
DHCP		Error	DHCP REBIND sent – No response for <P1><TAGS>	P1=IPv4 or IPv6			docsDevCmtsEventNotif
DHCP		Error	DHCP RENEW WARNING – Field invalid in response <P1> option<TAGS>	P1=v4			docsDevCmtsEventNotif
DHCP		Critical	DHCP RENEW FAILED - Critical field invalid in response				docsDevCmtsEventNotif
DHCP		Error	DHCP REBIND WARNING – Field invalid in response <TAGS>				docsDevCmtsEventNotif
DHCP		Critical	DHCP REBIND FAILED - Critical field invalid in response				docsDevCmtsEventNotif

Process	Sub-Process	CMTS/CCAP Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
DHCP		Notice	DHCP Reconfigure received<TAGS>				docsDevCmtsEventNotif
DHCP		Notice	DHCP Renew - lease parameters <P1> modified<TAGS>	P1 = list of params that changed at renew			docsDevCmtsEventNotif
DHCP		Error	Primary lease failed, IPv4 fallback initiated<TAGS>				docsDevCmtsEventNotif
DHCP		Critical	DHCP Failed - CCAP Core list missing				docsDevCmtsEventNotif
Init	DHCP	Critical	DHCP FAILED – Discover sent, no offer received<TAGS>				docsDevCmtsEventNotif
Init	DHCP	Critical	DHCP FAILED – Request sent, No response<TAGS>				docsDevCmtsEventNotif
Init	DHCP	Warning	DHCP WARNING - Non-critical field invalid in response <TAGS>				docsDevCmtsEventNotif
Init	DHCP	Critical	DHCP FAILED – Critical field invalid in response <TAGS>				docsDevCmtsEventNotif
Init	DHCP	Critical	DHCP failed – RS sent, no RA received<TAGS>				docsDevCmtsEventNotif
Init	DHCP	Critical	DHCP Failed – Invalid RA<TAGS>				docsDevCmtsEventNotif
Init	DHCP	Critical	DHCP failed – DHCP Solicit sent, No DHCP Advertise received<TAGS>				docsDevCmtsEventNotif
Init	DHCP	Critical	DHCP failed – DHCP Request sent, No DHCP REPLY received<TAGS>				docsDevCmtsEventNotif
Init	DHCP	Error	Primary address acquired, secondary failed<TAGS>				docsDevCmtsEventNotif
Init	DHCP	Error	Primary address failed, secondary active<TAGS>				docsDevCmtsEventNotif
Init	IPv6 Address Acquisition	Critical	Link-Local address failed DAD<TAGS>				docsDevCmtsEventNotif
Init	IPv6 Address Acquisition	Critical	DHCP lease address failed DAD<TAGS>				docsDevCmtsEventNotif
Init	TOD	Warning	ToD request sent – No Response received<TAGS>				docsDevCmtsEventNotif

Process	Sub-Process	CMTS/CCAP Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
Init	TOD	Warning	ToD Response received – Invalid data format<TAGS>				docsDevCmtsEventNotif
Init	802.1x Authentication						docsDevCmtsEventNotif
Init	IKE Mutual Authentication						docsDevCmtsEventNotif
TOD		Error	ToD request sent- No Response received<TAGS>				docsDevCmtsEventNotif
TOD		Error	ToD Response received – Invalid data format<TAGS>				docsDevCmtsEventNotif
Secure Software Download	**4**						
SW Upgrade	SW UPGRADE INIT	Notice	SW Download INIT – Via NMS	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address			docsDevCmtsEventNotif
SW Upgrade	SW UPGRADE INIT	Notice	SW Download INIT – Via GCP	Other than Local Log, append: SW file: <P2> - SW server: < P3><TAGS> P2 = SW file name P3 = SW Download server IP address			docsDevCmtsEventNotif
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	SW Upgrade Failed during download – Max retry exceed (3)	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address			docsDevCmtsEventNotif

Process	Sub-Process	CMTS/CCAP Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	SW Upgrade Failed Before Download – Server not Present	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address			docsDevCmtsEventNotif
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	SW upgrade Failed before download – File not Present	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address			docsDevCmtsEventNotif
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	SW upgrade Failed before download –TFTP Max Retry Exceeded	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address			docsDevCmtsEventNotif
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	SW upgrade Failed after download –Incompatible SW file	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address			docsDevCmtsEventNotif

Process	Sub-Process	CMTS/CCAP Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	SW upgrade Failed after download – SW File corruption	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address			docsDevCmtsEventNotif
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	Disruption during SW download – Power Failure	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address			docsDevCmtsEventNotif
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	Disruption during SW download – RF removed	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address			docsDevCmtsEventNotif
SW Upgrade	SW UPGRADE SUCCESS	Notice	SW download Successful – Via NMS	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address			docsDevCmtsEventNotif

Process	Sub-Process	CMTS/CCAP Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
SW Upgrade	SW UPGRADE SUCCESS	Notice	SW download Successful – Via Config file	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address			docsDevCmtsEventNotif
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	Improper Code File Controls	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address			docsDevCmtsEventNotif
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	Code File Manufacturer CVC Validation Failure	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address			docsDevCmtsEventNotif
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	Code File Manufacturer CVS Validation Failure	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address			docsDevCmtsEventNotif

Process	Sub-Process	CMTS/CCAP Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	Code File Co-Signer CVC Validation Failure	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address			docsDevCmtsEventNotif
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	Code File Co-Signer CVS Validation Failure	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address			docsDevCmtsEventNotif
SW Upgrade	VERIFICATION OF CVC	Error	Improper Configuration File CVC Format	Other than Local Log, append: Config file: <P1> - Config file server: < P2><TAGS> P1 = Config file name P2 = Config file server IP address			docsDevCmtsEventNotif
SW Upgrade	VERIFICATION OF CVC	Error	Configuration File CVC Validation Failure	Other than Local Log, append: Config file: <P1> - Config file server: < P2><TAGS> P1 = Config file name P2 = Config file server IP address			docsDevCmtsEventNotif

Process	Sub-Process	CMTS/CCAP Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
SW Upgrade	VERIFICATION OF CVC	Error	Improper SNMP CVC Format	Other than local Log, append: SNMP Manager: <P1><TAGS> P1= IP Address of SNMP Manager			docsDevCmtsEventNotif
SW Upgrade	VERIFICATION OF CVC	Error	SNMP CVC Validation Failure	Other than local Log, append: SNMP Manager: <P1><TAGS> P1= IP Address of SNMP Manager			docsDevCmtsEventNotif
Diagnostic Log	11						
Diag	LogSize	Warning	Diagnostic log size reached high threshold. Enabled detectors: <P1>;Log maximum size: <P2>	P1 = (ASCII hex representation of enabled diagnostic log detectors bit mask) P2 = maximum size of the diagnostic log			docsDevCmtsEventNotif
Diag	LogSize	Notice	Diagnostic log size dropped to low threshold. Enabled detectors: <P1>;Log maximum size: <P2>	P1 = (ASCII hex representation of enabled diagnostic log detectors bit mask) P2 = maximum size of the diagnostic log			docsDevCmtsEventNotif
Diag	LogSize	Warning	Diagnostic log size reached full threshold. Enabled detectors: <P1>;Log maximum size: <P2>	P1 = (ASCII hex representation of enabled diagnostic log detectors bit mask) P2 = maximum size of the diagnostic log			docsDevCmtsEventNotif
Physical and Environmental							
RPD- PE	Cooling	Warning	Cooling - Sensor unit=<P1> - High Temperature Threshold Exceeded <P2>	P1 = entPhysicalIndex of temperature sensor P2 = Temp (F/C)			docsDevCmtsEventNotif

Process	Sub-Process	CMTS/CCAP Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
RPD- PE	Cooling	Warning	Cooling - Sensor unit=<P1> - Normal Operating Temperature Exceeded: <P2>	P1 = entPhysicalIndex of temperature sensor P2 = Temp (F/C)			docsDevCmtsEventNotif
RPD- PE	Power	Warning	Power - Power supply - Below 95%				docsDevCmtsEventNotif
RPD- PE	Power	Critical	Power - Power Supply - Improper Input Voltage				docsDevCmtsEventNotif
RPD-PE	Security	Warning	Lid opened				docsDevCmtsEventNotif

Optical events? Loss of signal? Receive level below threshold? Laser temperature threshold?

B.1 Example SNMP Notification and Syslog Event Message (Informative)

The following is an example SNMP Notification and Syslog message "Event Message" text string for Event ID 70000304:

```
Power - Power Supply Bus Failure; unit=pw/1/1/;
```

Appendix I Sample CCAP XML Configuration (Informative)

I.1 CCAP XML Configuration File

Appendix II Future Updates (Informative)

NOTE: The following sections will be updated in a future release of this specification**:** 6.6, 7.4.2.4, 7.4.2.7, 8, 10, 11, 11.1.1, 11.1.2, 11.4.3.1, 11.4.5.3, 12, Annex B, and Appendix I.

Appendix III Acknowledgments (Informative)

On behalf of the cable industry and our member companies, CableLabs would like to thank the following individuals for their contributions to the development of this specification.

Contributor	Company Affiliation
Tom Ferreira	Arris
Niki Pantelias	Broadcom
Nikhil Tayal	CableLabs
Pawel Sowinski	Cisco
Joe Solomon	Comcast
John Bevilacqua	Comcast
Andrew Sundelin	Dial in the Sun, LLC
Michael Patrick	Harmonic, Inc.
Rei Brockket	Pace