Final draft ETSI ES 203 259 V1.1.1 (2014-11)

Integrated broadband cable telecommunication networks (CABLE); Converged Cable Access Platform; Operational Support System Interface

Reference
DES/CABLE-00010

Keywords
access, broadband, cable, DOCSIS, IPCable, modem

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This final draft ETSI Standard (ES) has been produced by ETSI Technical Committee Integrated broadband cable telecommunication networks (CABLE), and is now submitted for the ETSI standards Membership Approval Procedure.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**may not**", "**need**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document defines the requirements necessary for the Configuration, Fault Management, and Performance Management of the Converged Cable Access Platform (CCAP) system. The intent of the present document is to define a common, cross-vendor set of functionality for the configuration and management of CCAPs.

The present document defines a standard configuration object model for the configuration of the CCAP. The present document also defines the Simple Network Management Protocol (SNMP) Management requirements for a CCAP. These SNMP requirements include both protocol conformance and management object definitions, based largely upon existing industry standard management objects found in DOCSIS CMTSs (Cable Modem Termination Systems) and Universal EQAMs (Edge Quadrature Amplitude Modulation). In addition, the present document defines the standard Event Messaging requirements of a CCAP system.

The present document corresponds to the CableLabs CCAP-OSSI specification [i.21].

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are necessary for the application of the present document.

[1] DPoE-SP-OSSIv1.0-I05-130808 (August 8, 2013): "DOCSIS Provisioning of EPON OSSI Specifications, DPoE™ Operations and Support System Interface Specification", Cable Television Laboratories, Inc.

[2] DPoE-SP-OSSIv2.0-I03-130808 (August 8, 2013): "DOCSIS Provisioning of EPON OSSI Specification, DPoE™ Operations and Support System Interface Specification", Cable Television Laboratories, Inc.

[3] ETSI ES 203 385 (2014-xx): "CABLE; DOCSIS® Layer 2 Virtual Private Networking".

[4] CM-SP-M-OSSI-I08-081209 (December 9, 2008): "Data-Over-Cable Service Interface SpecificationsModular Headend Architecture, DOCSIS M-CMTS Operations Support Interface Specification", Cable Television Laboratories, Inc.

[5] ISO/IEC 13818-1 (2007): "Information technology -- Generic coding of moving pictures and associated audio information: Systems".

[6] ETSI EN 302 878-4 (V1.1.1) (2011-11): "Access, Terminals, Transmission and Multiplexing (ATTM); Third Generation Transmission Systems for Interactive Cable Television Services - IP Cable Modems; Part 4: MAC and Upper Layer Protocols; DOCSIS 3.0".

[7] CM-SP-OSSIv3.0-I21-130404 (April 4, 2013): "DOCSIS 3.0 Operations Support System Interface Specification", Cable Television Laboratories, Inc.

[8] IANA: "Service Name and Transport Protocol Port Number Registry".

NOTE: Available at: http://www.iana.org/assignments/port-numbers.

[9] IETF RFC 1350: "The TFTP Protocol (Revision 2)", July 1992.

[10] IETF RFC 2560: "X.509 Internet Public Key Infrastructure Online Certification Status Protocol - OCSP", June 1999.

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

[12] IETF RFC 3083: "Baseline Privacy Interface Management Information Base for DOCSIS Compliant Cable Modems and Cable Modem Termination Systems", March 2001.

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

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

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

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

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

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

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

[20] 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.

[21] 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.

[22] IETF RFC 4742: "Using the NETCONF Configuration Protocol over Secure Shell (SSH)", December 2006.

[23] IETF RFC 5246: "The Transport Layer Security (TLS) Protocol Version 1.2", August 2008.

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

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

[26] IETF RFC 6241: "Network Configuration Protocol (NETCONF)", June 2011.

[27] IETF RFC 6243: "With-defaults Capability for NETCONF", June, 2011.

[28] ANSI SCTE 154-2 2008: "SCTE-HMS-QAM-MIB".

[29] ANSI SCTE 154-4 2008: "MPEG Management Information Base - SCTE-HMS-MPEG-MIB".

[30] ANSI SCTE 154-5 2008: "SCTE-HMS-HEADENDIDENT Textual Conventions MIB".

2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] CM-TR-CCAP-V03-120511 (May 11, 2012): "Converged Cable Access Platform Architecture Technical Report", Cable Television Laboratories, Inc.

[i.2] ETSI EN 302 878-3 (V1.1.1) (11-2011): "Access, Terminals, Transmission and Multiplexing (ATTM); Third Generation Transmission Systems for Interactive Cable Television Services - IP Cable Modems; Part 3: Downstream Radio Frequency Interface; DOCSIS 3.0".

[i.3] DOCSIS Set-top Gateway (DSG) Interface Specification: "CM-SP-DSG", Cable Television Laboratories, Inc.

[i.4] Recommendation ITU-T M.3400 (02-2000): "TMN AND Network Maintenance: International Transmission Systems, Telephone Circuits, Telegraphy, Facsimile and Leased Circuits, TMN management functions".

[i.5] CM-SP-EQAM-PMI-I02-111117 (November 17, 2011): "Data-Over-Cable-Service-Interface Specifications, Modular Headend ArchitectureEdge QAM Provisioning and Management Interface Specification", Cable Television Laboratories, Inc.

[i.6] IETF RFC 1042: "Standard for the transmission of IP datagrams over IEEE 802 networks", February 1988.

[i.7] IETF RFC 1123: "Requirements for Internet Hosts - Application and Support", October 1989.

[i.8] IETF RFC 2326: "Real Time Streaming Protocol (RTSP)", April 1998.

[i.9] IETF RFC 3168: "The Addition of Explicit Congestion Notification (ECN) to IP", September 2001.

[i.10] IETF RFC 3260: "New Terminology and Clarifications for Diffserv", April 2002.

[i.11] IETF RFC 3414: "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", December 2002.

[i.12] IETF RFC 4001: "Textual Conventions for Internet Network Addresses", February 2005.

[i.13] IETF RFC 4131: "Management Information Base for Data Over Cable Service Interface Specification (DOCSIS) Cable Modems and Cable Modem Termination Systems for Baseline Privacy Plus", September 2005.

[i.14] IETF RFC 4743: "Using NETCONF over the Simple Object Access Protocol (SOAP)", December 2006.

[i.15] IETF RFC 4744: "Using the NETCONF Protocol over the Blocks Extensible Exchange Protocol (BEEP)", December 2006.

[i.16] IETF RFC 4291: "IP Version 6 Addressing Architecture", February 2006.

[i.17] IETF RFC 5519: "Multicast Group Membership Discovery MIB", April 2009.

[i.18] IETF RFC 5539: "NETCONF over Transport Layer Security (TLS)", May 2009.

[i.19] IETF RFC 6020: "YANG - A data modeling language for the Network Configuration Protocol (NETCONF)", October 2010.

[i.20] IETF RFC 6021: "Common YANG Data Types", October 2010.

[i.21] CM-SP-CCAP-OSSI-I05-130808 (August 8, 2013): "Converged Cable Access Platform Operational Support System Interface", Cable Television Laboratories, Inc.

[i.22] Recommendation ITU-T X.509: "Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

aggregation: special type of object association for Configuration Object Models in which objects are assembled or configured together to create a more complex object

Cable Modem (CM): modulator-demodulator at subscriber locations intended for use in conveying data communications on a cable television system

Cable Modem Termination System (CMTS: access-side networking element or set of elements that includes one or more MAC Domains and one or more Network System Interfaces

NOTE: 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.

Command Line Interface (CLI): 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

NOTE: The CCAP is configured by specifying the attributes of these objects.

Converged Cable Access Platform (CCAP): 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

Customer Premises Equipment: equipment at the end user's premises

NOTE: May be provided by the service provider.

datastore: collection of configuration objects used by the CCAP to define its configuration

Downstream: transmissions from CCAP to CM/CPE

NOTE: Also, RF spectrum used to transmit signals from a cable operator's headend or hub site to subscriber locations.

Edge QAM (EQAM): headend or hub device that receives packets of digital video or data

NOTE: It repacketizes the video or data into an MPEG transport stream and digitally modulates the digital transport stream onto a downstream RF carrier using quadrature amplitude modulation (QAM).

Extensible Markup Language: universal file format for storing and exchanging structured data

NOTE: 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.

flow: stream of packets used to transport data of a certain priority from the source to the sink

generalization: relationship in which one configuration model element (the child) is based on another model element (the parent)

NOTE: 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 (HFC): 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 (IEEE): voluntary organization which, among other things, sponsors standards committees and is accredited by the American National Standards Institute (ANSI)

Internet Engineering Task Force (IETF): body responsible for, among other things, developing standards used in the Internet

Internet Protocol (IP): internet network-layer protocol

Internet Protocol Detail Records (IPDR): Provides information about Internet Protocol (IP)-based service usage and other activities that can be used by Operational Support Systems (OSS) and Business Support Systems (BSS)

MAC Domain: grouping of Layer 2 devices that can communicate with each other without using bridging or routing

NOTE: In DOCSIS, it is the group of CMs that are using upstream and downstream channels linked together through a MAC forwarding entity.

management: functions on the CCAP that monitor for faults and for overall system performance, including traps and alarms

Media Access Control (MAC): used to refer to the Layer 2 element of the system

NOTE: This would include DOCSIS framing and signaling.

Management Information Base (MIB): database of device configuration and performance information which is acted upon by SNMP

Multimedia Terminal Adapter (MTA): combination cable modem and telephone adapter

Network Configuration Protocol: IETF network management protocol that provides mechanisms to manipulate the configuration of a device, commonly referred to as NETCONF

NOTE: NETCONF executes YANG-based XML files containing configuration objects.

pyang: YANG validator, transformer, and code generator, written in Python

NOTE: It is used to generate the CCAP schema file from the CCAP YANG modules.

Quadrature Amplitude Modulation (QAM): 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 (RF): in cable television systems, this refers to electromagnetic signals in the range 5 MHz to 1 000 MHz

Remote Authentication Dial In User Service (RADIUS): networking protocol that provides centralized authentication, authorization, and accounting (AAA) management for computers to connect and use a network service

Request for Comments (RFC): technical policy document of the IETF

NOTE: These documents can be accessed at http://www.rfc-editor.org/.

running-config: configuration objects that control CCAP behavior, along with any vendor-proprietary configurations

Secure Copy Protocol: secure file transfer protocol based on Secure Shell (SSH)

Simple Network Management Protocol (SNMP): allows a host to query modules for network-related statistics and error conditions

specialization: relationship in which one configuration model element (the parent) is used to model another element (the child)

NOTE: 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: configuration objects stored in non-volatile memory

Terminal Access Controller Access-Control System Plus (TACACS+): protocol that provides access control for routers, network access servers and other networked computing devices via one or more centralized servers

upstream: transmissions from CM to CCAP

NOTE: Also, RF spectrum used to transmit signals from a subscriber location to a cable operator's headend or hub site.

Video-on-Demand (VOD) System: system that enables individuals to select and watch video

X.509: Recommendation ITU-T standard for a public key infrastructure (PKI) for single sign-on (SSO) and Privilege Management Infrastructure (PMI)

YANG: data modeling language for the NETCONF network configuration protocol

NOTE: 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.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AAA	Authentication, Authorization, and Accounting
ABNF	Augmented Backus–Naur Form
ACL	Access Control List
ASN	Autonomous System Number
ATDMA	Adaptive TDMA
BPS	Bits Per Second
CA	Certificate Authority
CAS	Conditional Access System
CAT	Conditional Access Table
CCAP	Converged Cable Access Platform
CCI	Copy Control Information
CLI	Command Line Interface
CM	Cable Modem
CMTS	Cable Modem Termination System
CN	Common Name
COPS	Common Open Policy Service
CPAF	Configuration, Performance, Accounting, Fault
CPE	Customer Premises Equipment
CRL	Certificate Revocation List
CW	Control Word
DCD	Data Carrier Detect
DCS	Downstream Channel Sets
DHCP	Dynamic Host Configuration Protocol
DLC	Downstream Line Card
DNS	Domain Name Server
DOCSIS	Data-Over-Cable Service Interface Specification
DPoE	DOCSIS Provisioning of EPON
DS	Downstream
DSG	DOCSIS Settop Gateway
DSID	Downstream Service Identifier
DST	Daylight Saving Time
DTI	DOCSIS Timing Interface
DVB-C	Digital Video Broadcasting for Cable
EAE	Early Authentication and Encryption
ECM	Entitlement Control Message
ECMD	ECM Decoder
ECMG	ECM Generator
EMM	Entitlement Management Message
EPON	Ethernet Passive Optical Network
EQAM	Edge QAM
ERM	Edge Resource Manager
ERMI	Edge Resource Manager Interface
ERRP	Edge Resource Registration Protocol
ETV	Enhanced Television
FCAPS	Fault, Configuration, Accounting, Performance and Security
FEC	Forwarding Equivalence Class
FQDN	Fully Qualified Domain Name
FRU	Field Replaceable Unit
FTP	File Transfer Protocol
GCR	Group Classifier Rule
GLBG	General Load Balancing Group
GQC	Group QoS Configuration

GSF	Group Service Flow
HFC	Hybrid Fiber/Coax System
HMS	Hybrid Management Sub-Layer
HTTPS	Secure Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronic Engineers
IETF	Internet Engineering Task Force
IF-MIB	Interface MIB
IGMP	Internet Group Management Protocol
IP	Internet Protocol
IPDR	Internet Protocol Detail Records
ITU	International Telecommunication Union
LAN	Local Area Network
LSB	Least Significant Bit
MAC	Media Access Control
MAP	Bandwidth Allocation Map
MDD	MAC Downstream Descriptor
MGMD	Multicast Group Membership Discovery
MIB	Management Information Base
MIBS	Management Information Base
MLD	Multicast Listener Discovery
MPCP	Multi-Point Control Protocol
MPEG	Moving Picture Experts Group
MPTS	Multi-Program Transport Stream
MSB	Most Significant Bit
MTA	Multimedia Terminal Adapter
NA	Not Applicable
NETCONF	Network Configuration Protocol
NOC	Network Operations Center
NSI	Network Side Interface
NTP	Network Time Protocol
OAM	Object Access Method
OCSP	Online Certification Status Protocol
OID	Object Identifier
ONU	Optical Network Unit
OR	Logical OR
OSS	Operations Support System
OSSI	Operations Support System Interface
OUI	Organization Unique Identifier
PAT	Program Association Table
PCMM	IPCablecom Multimedia
PCR	Peak Cell Rate
PE	Physical & Environmental
PHS	Payload Header Suppression
PHY	Physical Layer
PID	Packet Identifier
PMI	Privilege Management Infrastructure
PMT	Program Map Table
PON	Passive Optical Network
QAM	Quadrature Amplitude Modulation
QOS	Quality of Service
RADIUS	Remote Authentication Dial In User Service
RC	Read-Create
RCC	Receive Channel Configuration
RCP	Receive Channel Profile
RF	Radio Frequency
RFC	Request for Comments
RKS	Record Keeping Server
RO	Read-Only
RPC	Remote Procedure Call
RSA	Remote Supervisor Adapter
RTSP	Real Time Streaming Protocol

RW	Read-Write
SA	Security Association
SAMIS	Subscriber Accounting Management Interface Specification
SAV	Source Address Verification
SCDMA	Synchronous Code Division Multiple Access
SCP	Secure Copy Protocol
SCTE	Society of Cable Telecommunications Engineers
SDV	Switched Digital Video
SNMP	Simple Network Management Protocol
SNTP	Simple Network Time Protocol
SRE	System Route Engine
SSH	Secure Socket Shell
SSM	Source Specific Multicast
TACACS+	Terminal Access Controller Access-Control System Plus
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TFTP	Trivial File Transfer Protocol
TLS	Transport Layer Security
TLV	Type Length Value Attribute
ToS	Terms of Service
TS	Transport Stream
TSID	Transport Stream Identifier
UCID	Upstream Channel Identifier
UDC	Upstream Drop Classifier
UDP	User Datagram Protocol
ULC	Upstream Line Card
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URN	Uniform Resource Name
US	Upstream
UTC	Coordinated Universal Time
UTF	Unicode Transformation Format
VIVSO	Vendor-Identifying Vendor Options
VOD	Video-On-Demand
WAN	Wide Area Network
XML	Extensible Markup Language
XSD	XML Schema Document

4 CCAP Data Reference Architecture

Figure 4.1, displays the interfaces used for the CCAP. The present document will focus on the Operations Support System Interface (OSSI) between the CCAP and the Operations Support System (OSS). The interfaces between the OSS and the eSAFE and Cable Modems are out of scope for the present document.

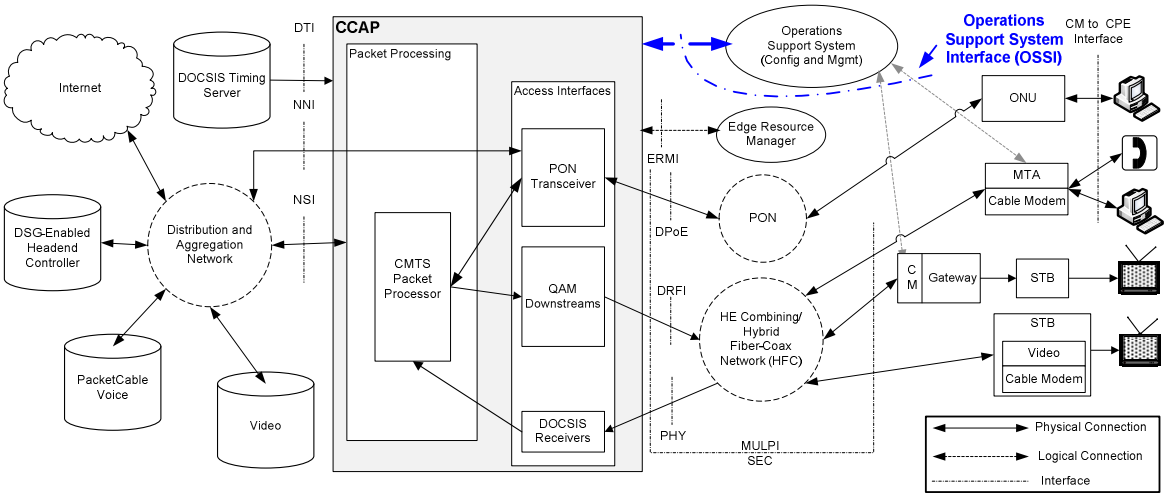


Figure 4.1: CCAP Interface Reference Architecture

Within a CCAP implementation, logical interface connectivity is from the OSSI to the packet processing function of the CCAP. This logical interface connection allows for the configuration and management of the CCAP infrastructure. The packet processing function will receive its OSS content through the Network Side Interface (NSI), consisting of at least 160 Gbps of data on one or more physical interfaces.

For additional information about the CCAP data reference architecture, see [i.1].

5 Overview

5.1 FCAPS Network Management Model

The International Telecommunication Union (ITU) Recommendation [i.4] defines a set of management categories, referred to as the Fault, Configuration, Accounting, Performance and Security (FCAPS) model, represented by the individual management categories of Fault, Configuration, Accounting, Performance and Security. Telecommunications operators, including cable operators, commonly use this model to manage large networks of devices. The present document uses these management categories to organize the requirements for the configuration and management of the CCAP platform.

Each of these management categories is discussed in the following clauses.

5.1.1 Fault Management

Fault management is a proactive and on-demand network management function that allows non-standard/abnormal operation on the network to be detected, diagnosed, and corrected. A typical use case involves network elements detecting service-impacting abnormalities; when detected, an autonomous event (often referred to as an alarm notification) is sent to the network operations center (NOC) to alert the cable operator of a possible fault condition in the network affecting a customer's service. Once the cable operator receives the event notification, further troubleshooting and diagnostics can be performed by the cable operator to correct the fault condition and restore the service to proper operation. Example Fault Management use cases are shown in figure 5.1.

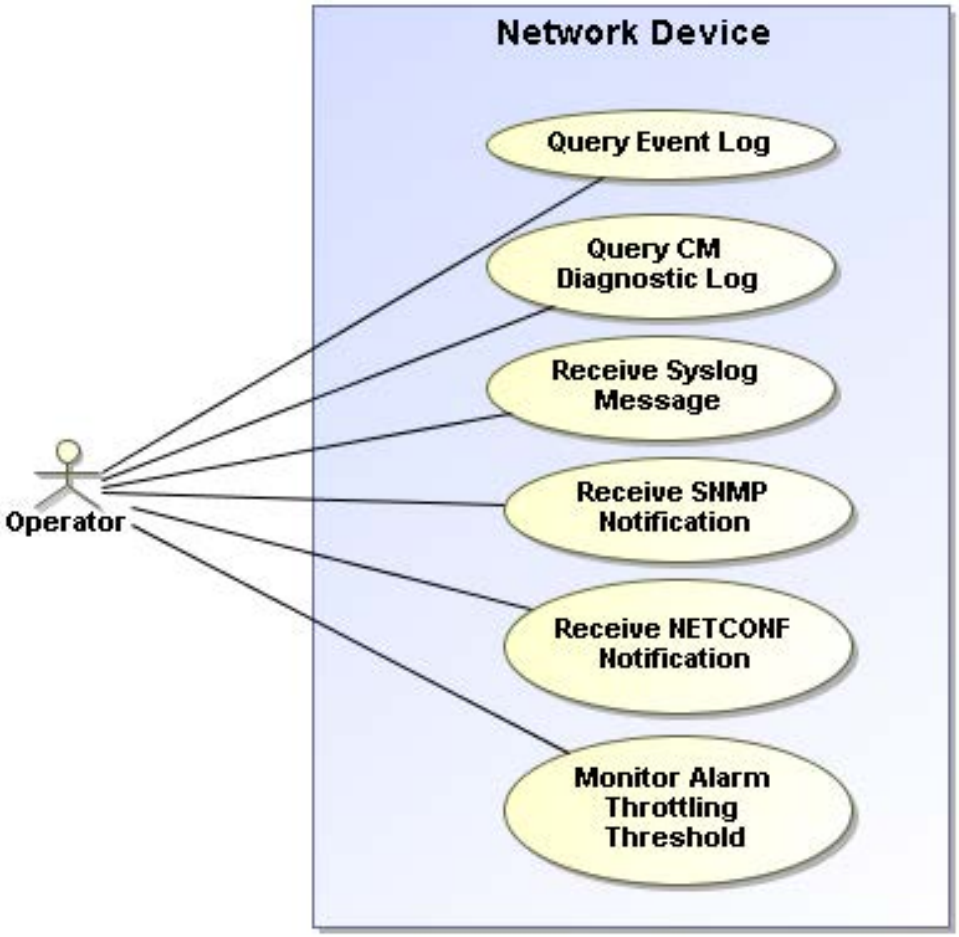


Figure 5.1: Fault Management Use Cases

5.1.2 Configuration Management

Configuration Management provides a set of network management functions that enables system configuration building and instantiating, installation and system turn up, network and device provisioning, auto-discovery, backup and restore, software download, status, and control (e.g. checking or changing the service state of an interface). Example Configuration Management use cases are shown in figure 5.2.

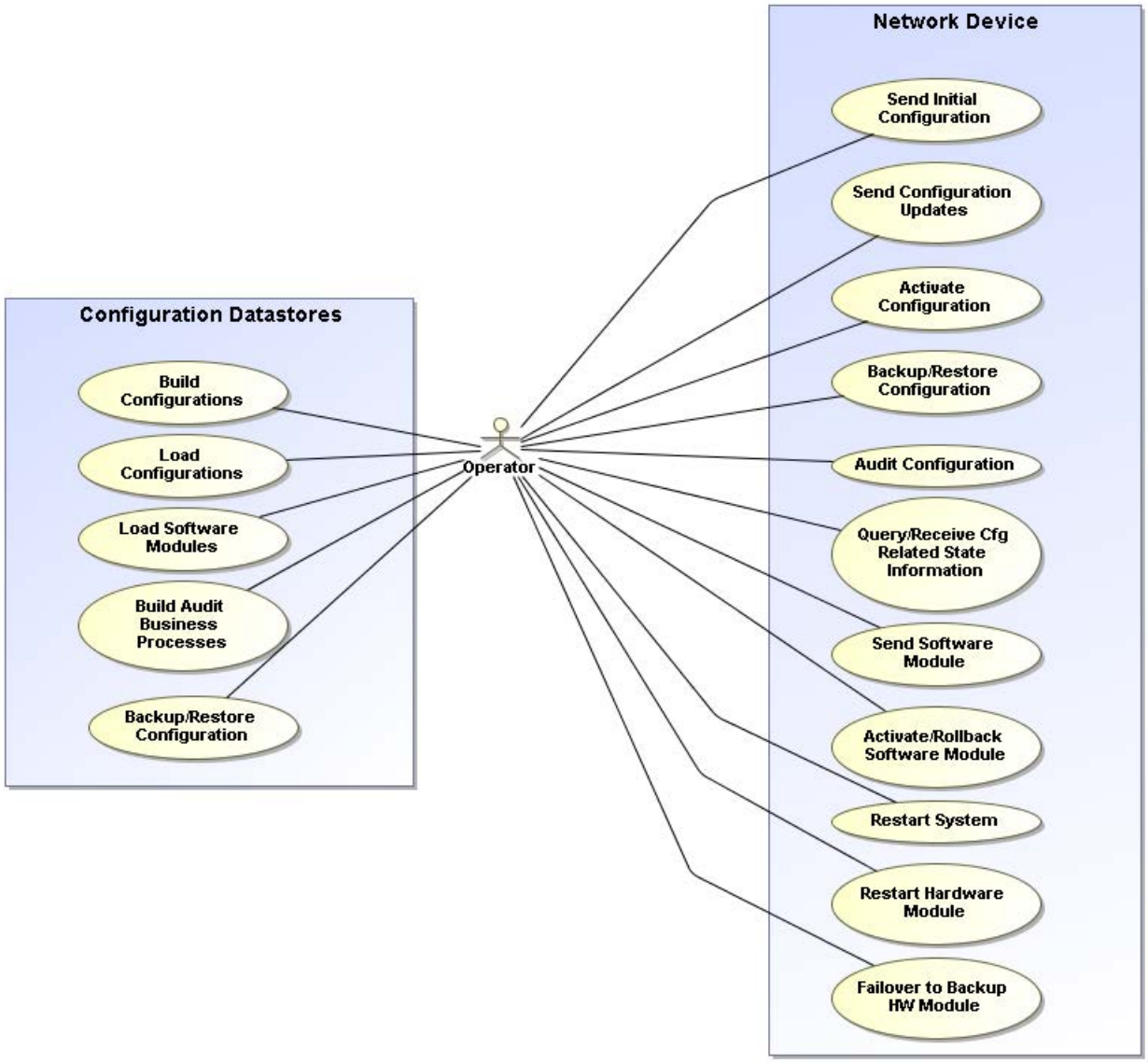


Figure 5.2: Configuration Management Use Cases

5.1.3 Accounting Management

Accounting Management is a network management function that allows cable operators to measure the use of network services by subscribers for the purposes of cost estimation and subscriber billing. Subscriber Accounting Management Interface Specification (SAMIS), as defined in DOCSIS [7], is an example of an implemented Accounting Management function.

5.1.4 Performance Management

Performance Management is a proactive and on-demand network management function. Recommendation ITU-T M.3400 [i.4] defines its role as gathering and analyzing "statistical data for the purpose of monitoring and correcting the behavior and effectiveness of the network, NEs, or other equipment and to aid in planning, provisioning, maintenance and the measurement of quality". A Performance Management use case might include the NOC performing periodic (15 min, for example) collections of QoS measurements from network elements to perform monitoring and identification of any potential performance issues that may be occurring with the service being monitored. With the historical data that has been collected, trending analysis can be performed to identify issues that may be related to certain times of day or other corollary events. The cable operator can run reports on the data to identify suspect problems in service quality, or the NOC application can be provisioned, so that when certain performance thresholds are violated, the cable operator is automatically notified that a potential service quality problem may be pending. Significant intelligence can be integrated into the NOC application to automate the ability to detect the possible degradation of a customer's service quality, and take actions to correct the condition. Service level agreement compliance is not possible without strong performance management.

5.1.5 Security Management

Security Management provides for the management of network and operator security, as well as providing an umbrella of security for the telecommunications management network functions. Security Management functions include authentication, access control, data confidentiality, data integrity, event detection, and reporting. A Security Management use case might include providing authentication and data confidentiality when transferring a configuration file that contains the entire configuration data set for the device to a network element. These functions are covered in context within the present document.

5.1.6 CCAP-OSSI Document Organization

The present document uses the FCAPS framework to group topics and content. In order to provide a more logical flow, one that mirrors processes in place at cable operators, 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 Configuration Management

6.1 CCAP Configuration Theory of Operation

The CCAP combines the functionality of an EQAM with a CMTS. While these are distinct functions, the configuration of the CCAP will treat these functions in a consolidated way. To facilitate the configuration of such a complex and dense device, the present document describes two methods for configuring the device:

- processing of an XML-based configuration file transferred to the device and executed locally;
- configuring the device via the NETCONF protocol.

Aspects of CCAP configuration include:

- standard data model for configuration, with vendor-specific extensions for the inclusion of proprietary features;
- XML-based configuration file;

- ability to configure a full set of standardized and vendor-proprietary configuration elements;
- ability to configure a partial set of standardized and vendor-proprietary configuration elements;
- light-weight protocols for transferring configuration files to and from the CCAP for XML-based file configuration;
- NETCONF options to manage the CCAP configuration.

It is anticipated that the CCAP will contain only basic default settings in its startup configuration when initially powered on, and the operator will begin configuration of the CCAP via serial console connection. Basic default settings are vendor-specific. The following sections define standardized CCAP configuration mechanisms and processes.

6.2 CCAP Configuration and Transport Protocol Requirements

6.2.1 Configuration Object Datastore

The CCAP shall implement the standard configuration objects defined by the present document.

These configuration objects control CCAP behavior and, along with any vendor-proprietary configurations, are referred to as the "running-config".

The CCAP shall provide a method for saving the state of the running-config to non-volatile memory. For NETCONF-based configuration, the NETCONF "copy-config" operation protocol provides this mechanism. However, for XML file-based configuration, this mechanism will be vendor-specific.

The configuration objects stored in non-volatile memory are referred to as the "startup-config".

6.2.2 RF Interface Configurability and Licensing

The CCAP shall support the configuration of the output frequency by direct configuration of the desired frequency.

The CCAP shall support configuring all RF interface parameters specified in the CCAP configuration object model.

6.2.3 DHCP Relay Agent Requirements

The CCAP shall support the configuration of the relay function of the Dynamic Host Configuration Protocol, as specified in EN 302 878-4 [6].

The CCAP shall support the ability to be configured with multiple concurrent DHCPv6 server addresses for routing mode operation.

If no DHCPv6 server addresses are configured on the CCAP, the CCAP should forward upstream DHCPv6 messages out of its network side interfaces to the DHCPv6 multicast group.

The CCAP shall support configuration of at least four distinct DHCP helper addresses, so that devices such as CMs, MTAs, and (Customer Premises Equipment) CPE can be directed to separate DHCP servers by a CCAP operating in non-routing mode.

The CCAP shall support the configuration of relay agent and VIVSO options. This does not imply that all DOCSIS features of the CCAP need to be governed by this setting.

The CCAP shall support the DHCPv6 VIVSO option for CM MAC address in RELAY-FORW. This is the equivalent of DHCPv4 option 82 remote-id for both CM and CPE.

The CCAP shall support the ability to configure the throttling rate of DHCP renews (unicast) to abate flooding of the DHCP server for routing mode operation.

6.2.4 Dynamic Management of QAMs

When the downloaded configuration file contains updates to the QAM channel parameter configuration, the CCAP can send an ERMI-1 UPDATE message with a Service Status indicating "maintenance mode" for the particular QAM channel(s) affected. Once there are no active dynamic sessions and no traffic on the static UDP ports for each QAM channel, the channel is taken down, updates made, and then brought up and advertised with a new UPDATE.

For details, refer to the EQAM Dynamic Provisioning section of [i.5].

There can be a minimum number of preconfigured QAMs for DOCSIS and an additional set of channels that are demand based. When demand is low, those additional channels are available for other services. Conversely, when demand is high, more of these resources are assigned to DOCSIS, up to a configurable limit.

6.2.4.1 Dynamic Assignment of SDV/VOD QAMs

The ERM will control QAMs for SDV and VOD.

6.2.5 Video Configuration Requirements

The CCAP shall enable adjustability of the size of the de-jitter buffer.

6.2.6 DOCSIS® Configuration Requirements

The CCAP shall support the configuration of blocked bandwidth limits.

The CCAP shall support the ability to configure Concatenation (configurable on/off for each MAC domain, for pre-DOCSIS 3.0 modems, MAC wildcard).

The CCAP shall support the ability to configure Fragmentation (configurable on/off for each MAC domain, for pre-DOCSIS 3.0 modems, MAC wildcard).

The CCAP shall support the ability to configure enabling/disabling IPv6 provisioning mode via the DOCSIS MDD message.

The CCAP shall support the ability to configure steering a modem to a different CMTS or CCAP based on TLV 43.11 - Service Type Identifier per EN 302 878-4 [6].

6.3 CCAP XML File-Based Configuration

6.3.1 CCAP XML Configuration File Theory of Operation

The CCAP will be configurable via the execution of an XML-based configuration file that holds the configuration details for the platform. The XML configuration files are conformant to the XML schemas based on the CCAP configuration object model specified in the present document. A given XML configuration file for the CCAP platform is expected to be validated against these schemas.

The use case for configuring a CCAP with an XML configuration file is depicted in figure 6.1.

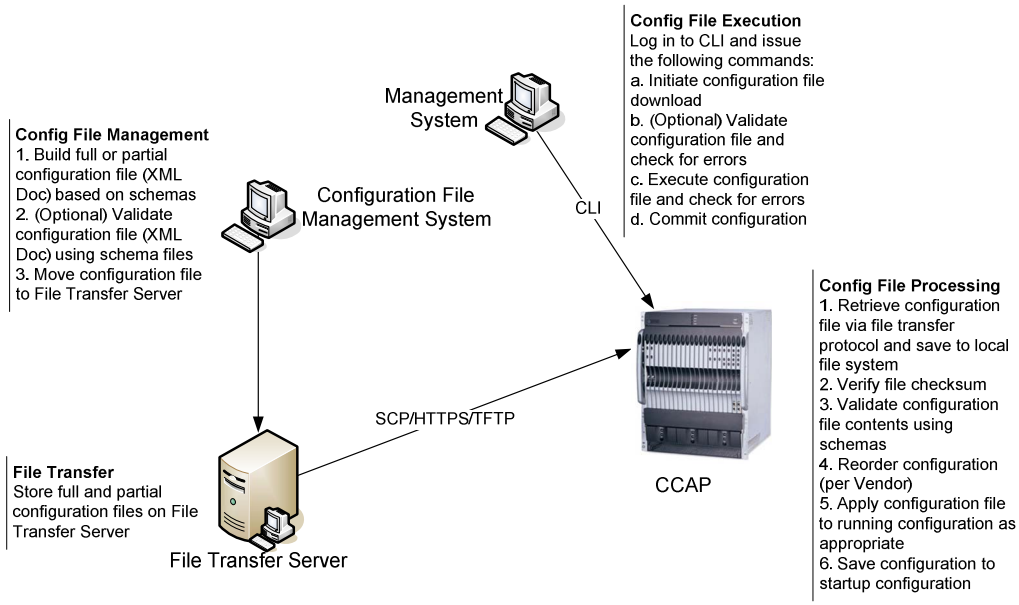


Figure 6.1: CCAP XML File-Based Configuration Use Case

The CCAP parses and processes XML configuration files that are stored locally. These files are generated by processes and systems out of scope for the present document. Operators place XML configuration files in CCAP local storage via file transfer. Before executing an XML configuration file, the CCAP verifies that it has not been corrupted in the file transfer process. The XML configuration file is then validated against the configuration file schema to ensure that the configuration is valid. This validation step can also be performed independent of configuration file execution.

The CCAP parses the entire XML configuration file and processes the configuration objects represented in the file in a vendor-proprietary manner. The CCAP allows partial execution of configuration files; invalid configuration instructions can be ignored while valid instructions will still be processed. The CCAP can also reject configuration instructions if they cannot be met by the capabilities of the hardware present.

The CCAP XML configuration file process is defined in the following clauses.

6.3.2 CCAP XML Configuration Files

The CCAP shall support the use of an XML configuration file to set the values of standard CCAP configuration objects. A CCAP XML configuration file is an XML UTF-8 text representation of the keysets and attribute values of a set of configuration objects.

The CCAP shall support the processing of an XML configuration file, which includes configuration elements that are related to hardware that is not installed in the CCAP chassis at the time of processing to allow for pre-provisioning.

Many configuration objects in the CCAP XML configuration file are nested within other configuration objects. The configuration file can contain the full hierarchy of elements where all element contexts are explicit. For an example of a configuration in a nested hierarchy, see clause H.1 CCAP XML Configuration File.

The present document makes use of the unified modeling language (UML) to define the common configuration elements of a CCAP. XML schemas and YANG modules are based on the CCAP configuration UML object model.

The CCAP shall support an XML configuration file that is conformant to the most recent version of the following XML schema:

```
ccap@yyyy-mm-dd.xsd
```

where yyyy-mm-dd represents the date on which the most recent version of the schema was published.

This schema is available at the following location: http://www.cablelabs.com/wp-content/uploads/specdocs/CCAP-OSSI-I05-schemas.zip.

A single XML configuration file - containing both standard as well as vendor-proprietary elements - will be delivered to the CCAP. The file makes use of both the standard and vendor-proprietary namespaces. An XML configuration file may include proprietary extensions targeting multiple vendors. While validating or executing the configuration file, the CCAP is expected to ignore proprietary extensions it does not support.

The CCAP shall only accept an XML configuration file if it indicates the version number(s) of the schemas/modules for which the file is intended to be conformant.

6.3.3 XML Configuration File Checksum

When the XML configuration file is downloaded or uploaded between the CCAP and a remote host, there exists a possibility that contents of the file may get corrupted or lost during a transfer.

The CCAP shall provide a POSIX-compliant MD5 "checksum" command used to verify the integrity of a downloaded XML configuration file.

The operator can compare the output of the checksum command for the file on the CCAP with that of a checksum command on the remote host where the configuration file originated to confirm that the file has not been altered or corrupted.

6.3.4 XML Configuration File Validation

Before attempting to execute a given XML configuration file on a CCAP, the operator might want to first validate the file against the XML schemas supported.

On a CCAP, it is anticipated that the operator will attempt to validate an XML configuration file that contains elements belonging to schemas of a single CCAP vendor. Note that even when an XML configuration file is successfully validated, errors could still be encountered when the same file is later executed by the operator.

The CCAP shall support a CLI command to validate an XML configuration file located on a local file system against the XML schemas supported by the software running on the CCAP.

The CCAP validate command shall validate that the XML configuration file is well-formed XML.

The CCAP validate command should also check the XML configuration file for the following:

- references to undefined configuration data;
- attribute value constraints;
- resource constraints.

When an XML configuration file successfully validates, the CCAP shall log an event with severity level "Info" (Event ID: 70000103).

When the validation of the XML configuration file experiences errors, the CCAP shall create a validation output log file to be stored on a local file system.

The CCAP shall name the validation output log file such that the name contains the configuration file name (e.g. <XML configuration file name>-<user>-<time>-validate.log.) and is different than the filename of the execution output log file defined in clause 6.3.6.

When the validation of the XML configuration file experiences any error, the CCAP shall include the following fields for each error entry, separated by a semi-colon (;), in the validation output log file:

- line number of the error;
- configuration element, including namespace;
- error message.

When an XML configuration file fails to validate, the CCAP shall provide an error message to the user via the user interface (regardless of the type of terminal session in use by the user), log an event with severity level "Notice" (Event ID: 70000102), and log the errored lines to the validation output log file as defined above.

6.3.5 XML Configuration File Execution Command and NETCONF Operations

Since the XML Configuration File downloaded to the CCAP is not automatically executed by the CCAP, it is necessary to define a CCAP CLI command to perform specific parsing and execution actions on a given XML Configuration File.

The CCAP shall support a CLI command to execute an XML configuration file located on a local file system, where the CCAP executes the operations specified for each element in the file.

The CCAP shall support the "merge", "replace", and "delete" operations defined in section 7.2 of [26]. The present document does not intend to make use of the "create" operation.

All configuration changes of an XML file are conceptually executed simultaneously, without regard to the order of the individual object operations in the file. The actual execution of an XML configuration file is expected to be implemented as a sequence of individual element operations in a vendor-specific order. Individual element operations can succeed or fail; the CCAP will log unsuccessful element operations.

The CCAP shall not reject a configuration object because it is dependent upon or related to a configuration object that occurs later in the configuration file and has not yet been processed.

For example, it is valid to execute an XML configuration file that contains an object_A that refers to a new object_B, when the object_A reference appears earlier in the file than the creation of object_B.

A "Full" XML configuration file is one that is intended to replace the entire set of configuration objects on the CCAP.

A Full XML configuration file will contain the operation="replace" attribute in the <ccap:ccap> tag at the root of the configuration tree. When the CCAP saves or exports the current running-config or the startup-config to an XML format, the CCAP shall insert operation="replace" in a single top-level <ccap:ccap> tag.

A "Partial" XML configuration file is one that is intended to augment the current running-config, replace a subset of the configuration objects on the CCAP, or to act on "control" objects (such as objects that allow a log to be reset or a diagnostic mode to be enabled).

A Partial XML configuration file will contain all of the parent object containers for the objects being configured, all the way up the configuration hierarchy to the "ccap" container. Because of this, caution is to be taken when using the "replace" or "delete" operations. While a "merge" operation will only update the attributes that are explicitly provided in the XML configuration file, the "replace" and "delete" operations act upon all objects within the configuration tree. This could cause the entire device configuration to be deleted.

A Partial XML configuration file using a merge or delete operation may exclude mandatory configurable attributes if they are not a key for the configuration object being acted upon. When a partial configuration is using a merge or delete operation, the CCAP shall ignore validation errors related to missing mandatory configuration attributes unless the missing attribute is the key for the configuration object being acted upon.

The following is an example of how to use the merge operation to update the configuration of a QAM channel on an existing downstream RF port. The file would have the following structure:

```
<ccap:ccap xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" SchemaVersion="2013-04-04"
xsi:schemaLocation="urn:cablelabs:params:xml:ns:yang:ccap ccap@2013-04-04.xsd"
xmlns:ccap="urn:cablelabs:params:xml:ns:yang:ccap" operation="merge">
  <chassis>
    <slot>
      <slot-number>1</slot-number>
      <rf-line-card>
          <ds-rf-port>
            <port-number>1</port-number>
            <down-channel>
              <channel-index>1</channel-index>
              <admin-state>up</admin-state>
              <power-adjust>4096</power-adjust>
              <frequency>200</frequency>
              <rf-mute>false</rf-mute>
              <qam-alias>tnt</qam-alias>
              <errp-advertising>false</ errp-advertising >
            </down-channel>
          </ds-rf-port>
      </rf-line-card>
    </slot>
```

```
  </chassis>
</ccap:ccap>
```

NOTE: When performing a merge or delete operation on a partial configuration file, only the attributes that define which instance is being configured are required; in the previous example the following attributes had to be specified:

- slot-number attribute of the slot object;
- port-number attribute of the downstream-port object;
- channel-index attribute of the down-channel object.

Example XML for Replacing a DownChannel object:

Note that all mandatory attributes of objects in the configuration tree are required when using a replace function; in this example a down-channel object is being replaced, but the mandatory attributes of line-card and downstream-port have to be included, even though they were already configured.

```
<ccap:ccap xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" SchemaVersion="2013-04-04"
xsi:schemaLocation="urn:cablelabs:params:xml:ns:yang:ccap ccap@2013-04-4.xsd"
xmlns:ccap="urn:cablelabs:params:xml:ns:yang:ccap" operation="replace">
  <chassis>
    <slot>
      <slot-number>1</slot-number>
      <rf-line-card>
        <rf-card>
          <line-card-name>DS RF 1</line-card-name>
          <admin-state>up</admin-state>
        </rf-card>
          <ds-rf-port>
            <port-number>1</port-number>
            <admin-state>up</admin-state>
            <down-channel>
              <channel-index>1</channel-index>
              <admin-state>down</admin-state>
              <power-adjust>0</power-adjust>
              <frequency>0</frequency>
              <rf-mute>false</rf-mute>
              <qam-alias>String</qam-alias>
              <errp-advertising>true</errp-advertising>
            </down-channel>
          </ds-rf-port>
      </rf-line-card>
    </slot>
  </chassis>
</ccap:ccap>
```

Example XML for Deleting a DownChannel Object:

```
<ccap:ccap xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" SchemaVersion="2013-04-04"
xsi:schemaLocation="urn:cablelabs:params:xml:ns:yang:ccap ccap@2013-04-04.xsd"
xmlns:ccap="urn:cablelabs:params:xml:ns:yang:ccap" operation="delete">
  <chassis>
    <slot>
      <slot-number>1</slot-number>
      <rf-line-card>
          <ds-rf-port>
            <port-number>1</port-number>
            <down-channel>
              <channel-index>100</channel-index>
            </down-channel>
          </ds-rf-port>
      </rf-line-card>
    </slot>
  </chassis>
</ccap:ccap>
```

The CCAP will support only one occurrence of the same NETCONF operation ("merge", "replace" or "delete") in a single file. If an operator needs to perform more than one operation type to configure the CCAP, separate configuration files will need to be created and the operator will execute those files sequentially.

If an XML configuration file does not contain one and only one explicit operation type value of "merge", "replace", or "delete", upon attempted execution of the file, the CCAP shall reject the entire file, make no changes to the running-config, and log the fatal error as an event with a severity level "Warning" (Event ID: 70000108).

If the ccap:ccap node in the XML configuration file has a "replace" operation value, and the subtree in the XML configuration file for that node is missing one or more mandatory elements (either standard elements or vendor-extensions), then the CCAP shall retain the mandatory elements, attempt the execution of the remaining elements in the file, and log the non-fatal errors as an event with a severity level "Error" (Event ID: 70000107).

The CCAP shall allow the pre-provisioning of configuration objects associated with line cards that are not yet present in the chassis.

Note that if a "replace" operation value is used, and parts of the subtree in the XML configuration file are missing one or more non-mandatory (vendor-specific or standard) elements, then the CCAP deletes the absent non-mandatory elements from its running-config.

Conversely, for a "merge" operation, the CCAP shall preserve both standard and vendor-extension objects in the affected subtree that are not present in the merged XML configuration file.

For a "delete" operation, the CCAP shall delete both standard and vendor-extension objects in the affected subtree.

If an XML configuration file contains an explicit "create" operation value, upon attempted execution of the file the CCAP shall reject the entire file, make no changes to the running-config, and log the fatal error as an event with a severity level of "Warning" (Event ID: 70000108).

6.3.6 XML Configuration File Parsing and Error Logging

Before a configuration file is applied to the CCAP, the CCAP performs several checks against the file. If the configuration file does not pass these checks, the CCAP will reject the file. The CCAP can also reject individual objects within the configuration file. In all rejection cases, the CCAP will log the rejection as an error.

When executed, the CCAP shall verify that the configuration file is well-formed XML.

If the CCAP fails to verify the file is well-formed as part of an execution, the CCAP shall reject the file, make no changes to the running-config, log the fatal error as an event with a severity level "Warning" (Event ID: 70000109), log the errored lines to the execution output log file in the format defined later in this section, and provide an error message to the user interface, regardless of the type of terminal session in use by the user.

If the module/schema version number contained within the executed XML configuration file is not compatible with the module/schema set supported by the CCAP, the CCAP shall reject the file, make no changes to the running-config, log the fatal error as an event with a severity level "Warning" (Event ID: 70000109), log the line(s) where the module/schema version mismatch was detected to the execution output log file in the format defined later in this section, and provide an error message to the user interface, regardless of the type of terminal session in use by the user.

When the execution of the XML configuration file completes without error, the CCAP shall log an event with severity level "Notice" (Event ID: 70000105).

When the execution of the XML configuration file completes without error, the CCAP may create an execution output log file containing execution time, user, and XML configuration filename information.

The CCAP supports "partial execution" of an XML configuration file, where certain elements in the file are successfully executed and other elements in the file are unable to be executed.

When the execution of the XML configuration file experiences errors, the CCAP shall create an execution output log file to be stored on a local file system.

The CCAP shall name the execution output log file such that the name contains the executed configuration file (e.g. <executed XML configuration file name>-<user>-<time>-out.log) and is different than the filename of the validation output log file defined in clause 6.3.4.

When the execution of the XML configuration file experiences any error, the CCAP shall use a standard format for each error entry, separated by a semi-colon (;), in the execution output log file and include:

- line number of the error;

- configuration element, including namespace;
- error message.

When an executed XML configuration file contains elements that are not supported by the CCAP, the CCAP shall process the elements it does support, and log the non-fatal error as an event with severity level "Error" (Event ID: 70000106), and log the unsupported lines to the execution output log file as defined above.

The CCAP shall perform the validate function as an initial step of the execute command before any changes are applied to the configuration store.

If the CCAP fails to validate the file as part of an execution, the CCAP shall reject the file, make no changes to the running-config, log the fatal error as an event with a severity level "Warning" (Event ID: 70000109), log the errored lines to the execution output log file as defined above, and provide an error message to the user interface, regardless of the type of terminal session in use by the user.

If during the execution of a validated configuration file an error is encountered, the CCAP shall apply the configuration of the non-errored elements, log the non-fatal error as an event with severity level "Error" (Event ID: 70000107), log the errored lines to the execution output log file as defined above, and provide an error message to the user interface, regardless of the type of terminal session in use by the user.

6.3.7 File Transfer Mechanisms

The CCAP will implement several file transfer mechanisms that can be used to "download" an XML configuration file or software image from an external host to the CCAP or "upload" a copy of an XML configuration file or software image to an external host.

The CCAP shall support the Secure Copy Protocol (SCP) - based on Secure Shell version 2 - for both file download and upload operations.

The CCAP shall support the initiation of Secure Copy download and upload operations from both a remote host and from the CCAP CLI.

The CCAP shall support the Trivial File Transfer Protocol (TFTP), as specified in RFC 1350 [9], for both file download and upload operations.

Since TFTP has no inherent authentication mechanism, the CCAP shall only support the initiation of Trivial File Transfer download and upload operations from the CCAP CLI by an authenticated and authorized user.

The CCAP should support Secure Hypertext Transfer Protocol (HTTPS) for both file download and upload operations.

The CCAP should support the initiation of HTTPS download and upload operations from both a remote host and from the CCAP CLI.

If HTTPS download initiation from a remote host is supported by the CCAP, the CCAP shall implement TLS validation of the X.509 certificate presented by the remote host.

For both SCP and HTTPS file download and upload operations, the CCAP shall support the ability to authenticate the file transfer connection via TACACS+ and RADIUS as well as usernames configured locally on the CCAP.

If an initiated file transfer fails, the CCAP shall log an event with severity level "Error" (Event ID: 70000102) and provide an error message to the user interface indicating that the file transfer failed, regardless of the type of terminal session in use by the user.

6.3.7.1 TLS for HTTPS

Authentication of the remote host server by the CCAP is performed by validating the certificate provided by the remote host during TLS setup.

The CCAP shall negotiate TLS-related integrity protection and encryption features at the TLS layer.

The remote host will always offer TLS cipher suites to be used for the session, as specified in RFC 5246 [23].

The CCAP shall decide which TLS cipher suites are used, as specified in RFC 5246 [23].

The CCAP shall verify that the data is sent and received according to RFC 5246 [23]. This verification is also used to detect if the received data has been tampered with.

The CCAP shall support the following TLS profiles (per RFC 5246 [23]):

- TLS_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA

The use of NULL integrity protection and/or NULL encryption by the CCAP is not anticipated.

The remote host will present X.509 digital certificates, per RFC 5280 [25], for authentication in TLS, as profiled in table 6.1.

Table 6.1: TLS Certificate Profile

TLS Server Certificates	
Subject Name Form	C=<Country> O=<Company> CN=<FQDN> FQDN is the remote host's fully qualified domain name (e.g. es.example.com). Only a single FQDN is allowed in the CN field. Additional fields may be present in the subject name.
Intended Usage	These certificates are used to authenticate TLS handshake exchanges (and encrypt when using RSA key exchange).
Validity Period	Set by operator policy RFC 5246 [23]
Modulus Length	1 024, 1 536, 2 048
Extensions	KeyUsage[critical](digitalSignature, keyEncipherment) RFC 5246 [23] extendedKeyUsage (id-kp-serverAuth, id-kp-clientAuth) authorityKeyIdentifier (keyIdentifier=<subjectKeyIdentifier value from CA cert>)

Remote host certificates will be issued by the cable operator.

The CCAP shall verify that the remote host's TLS certificates are part of a certificate chain that chains up to the cable operator's certificate authority (CA).

If changes other than the certificate serial number, validity period and the value of the signature exist in the root certificate that was sent by the remote host to the CCAP in comparison to the known root certificate, the CCAP shall conclude that the certificate verification has failed.

The CCAP shall build the certificate chain and validate the TLS certificate according to the "Certificate Path Validation" procedures described in RFC 5280 [25].

6.3.8 Exporting the Configuration Object Data Store

The CCAP shall support a CLI command to export the startup-config into an XML configuration file to be stored on local non-volatile storage.

The CCAP shall support a CLI command to export the current running-config of the device to an XML Configuration File to be stored on local non-volatile storage.

The CCAP shall support the export of XML configuration files in a format that conforms to the standard CCAP schema set, including optional vendor extensions.

The CCAP shall allow the user to specify the full file system path and filename of the exported XML Configuration File.

The CCAP should support XML configuration file export operations with both concise and verbose options. The output of the concise export operation does not include optional attributes that are set to the default value/have not been configured. The output of the verbose operation does include these items.

When exporting to an XML configuration file, the CCAP shall encrypt in a vendor-specific way the content of configuration items intended to be "secret", including, but not limited to:

- passwords (including lawful intercept);
- DOCSIS shared secret;
- TACACS+ and RADIUS keys;
- routing protocol keys.

The CCAP shall be able to import a previously exported configuration file containing encrypted attributes, where the configuration file was previously exported from that vendor's CCAP devices.

It is expected that encrypted attributes in an exported XML configuration file can be imported on another CCAP produced by the same vendor.

6.4 CCAP NETCONF-Based Configuration

6.4.1 NETCONF Theory of Operation

The CCAP may also support configuration via the NETCONF protocol. In this case configuration instructions are sent using XML-encoded remote procedure calls (RPCs) in NETCONF protocol messages from a configuration management tool to the CCAP. The XML configuration data, representing the CCAP configuration, is conformant to the YANG modules specified in the present document.

The use case for configuring a CCAP via NETCONF is depicted in figure 6.2.

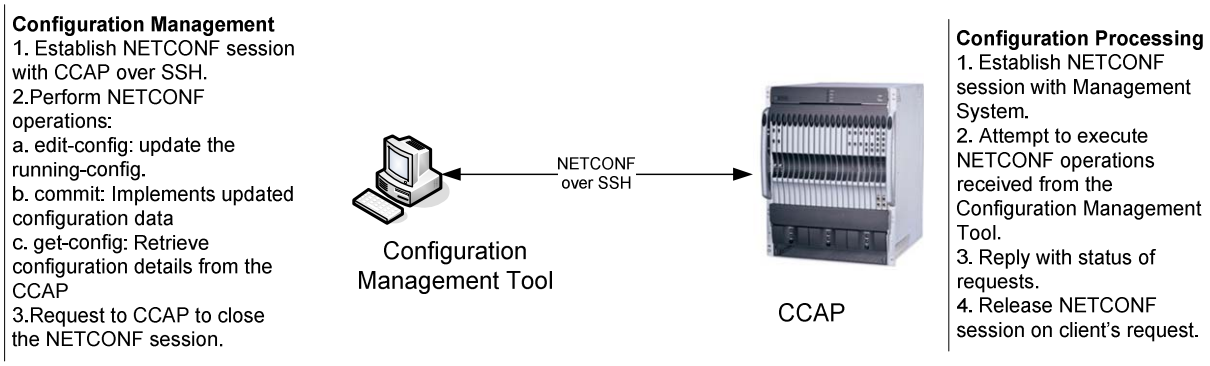


Figure 6.2: CCAP NETCONF-Based Configuration Use Case

The YANG modules, based on the CCAP configuration object model, are implemented by the configuration management tool and the CCAP; these modules are used to generate valid configuration NETCONF operations and content from the management system and to validate and execute those operations and content on the CCAP.

When the configuration management tool begins the configuration process, an SSH session is set up between the configuration management tool and the CCAP being configured. The configuration management tool can then deliver full or partial CCAP configuration changes using NETCONF operations. The configuration content can be machine-generated or hand created; they are sent in the NETCONF RPC to the CCAP.

The CCAP receives, parses, and processes the configuration operations received via NETCONF from the configuration management tool. The CCAP can be fully or partially reconfigured; invalid configuration instructions can be ignored while valid instructions will still be processed. The CCAP can also reject configuration instructions if they cannot be met by the capabilities of the hardware present.

The CCAP can also respond to <get-config> operations from the configuration management tool and provide a full or partial XML-based representation of the current device configuration, delivered to the configuration management tool via NETCONF.

The CCAP NETCONF configuration process is discussed in the following sections.

6.4.2 NETCONF Overview

NETCONF [26] is a configuration management protocol defined by the IETF. NETCONF provides mechanisms to install, manipulate, and delete the configuration of network devices.

NETCONF uses an XML-based data encoding for the configuration data as well as protocol messages. The protocol operations are realized on top of a simple Remote Procedure Call (RPC) layer. A client encodes an RPC in XML and sends it to a server using a secure, connection-oriented session. The server responds with a reply encoded in XML. The contents of both the request and the response are fully described using YANG ([i.19]) allowing both parties to recognize the syntax constraints imposed on the exchange.

NETCONF is connection-oriented, requiring a persistent connection between peers. This connection needs to provide reliable, sequenced data delivery. NETCONF connections are long-lived, persisting between protocol operations; the connection is also expected to provide authentication, data integrity, and confidentiality.

There are currently several transport mappings published, including SSHv2 RFC 4742 [22], SOAP, RFC 4743 [i.14], BEEP, RFC 4744 [i.15], and TLS, RFC 5539 [i.18]. The SSH transport protocol mapping is mandatory to implement and the others are optional.

In addition to the XML file based configuration of the CCAP, it is expected that some vendors will provide a NETCONF option for configuring and managing a CCAP using the YANG module specified in Annex G, YANG Configuration Module. These modules are based on the CCAP configuration object model specified in clause 6.5 UML Configuration Object Model.

6.4.3 NETCONF Requirements

The CCAP should implement NETCONF Server, as specified in RFC 6241 [26].

If the CCAP implements NETCONF Sever, the base NETCONF capability identified by the urn:ietf:params:netconf:base:1.0 URN shall be implemented; all remaining capabilities described in the RFC are optional.

Any NETCONF Server implemented in the CCAP shall comply with the mandatory SSHv2 transport mapping specified in RFC 4742 [22].

If the CCAP supports NETCONF-based configuration, the CCAP shall support the "merge", "replace", and "delete" operations defined in section 7.2 of RFC 6241 [26]. The present document does not intend to make use of the "create" operation.

If the CCAP supports NETCONF-based configuration, the CCAP shall support the ccap.yang module defined in Annex G, YANG Configuration Module, RFC 4743 [i.14]

If the CCAP supports NETCONF-based configuration, the CCAP should support the with-defaults Capability for NETCONF according to the 'report-all' basic mode, as defined in RFC 6243 [27]. A server that uses the 'report-all' basic mode does not consider any data node to be default data, even schema default data. If the CCAP supports NETCONF-based configuration, when a client retrieves data with a <with-defaults> parameter equal to 'report-all', the CCAP shall report all data nodes, including any data nodes considered to be default data by the server. This is the equivalent of a "verbose" XML configuration file export.

6.5 UML Configuration Object Model

6.5.1 CCAP UML Configuration Object Model Overview

The CCAP UML configuration object model, as well as the schemas based on that object model, have been 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.

The CCAP configuration object model strives to make maximum re-use of existing SCTE HMS and OSSIv3.0 MIBS and object models. In some cases these models were modified to address specific issues that were CCAP-related.

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

6.5.1.1 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 shall provide 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.

6.5.1.2 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 the present document 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.

6.5.1.3 Use of Interface Names in Configuration

Several configuration objects defined in the present document 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 the present document 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 should reject a configuration that includes an interface name that does not follow the vendor's naming conventions.

6.5.1.4 Unconstrained Strings in the Configuration Object Model

For object attributes with a data type of String, there are cases where the present document does not provide a length constraint. For these attributes, the CCAP may impose a vendor-specific length constraint. If a value in the XML configuration file exceeds this vendor-specific length constraint, the CCAP should truncate the text string to that limit. In addition, if a value in the XML configuration file exceeds this vendor-specific length constraint, the CCAP shall log the non-fatal error as an event with severity level "Error" (Event ID: 70000107), log the errored lines to the execution output log file, and provide an error message that describes the vendor-specific length constraint and details how the string was handled (truncated, rejected, etc.).

6.5.2 Vendor-Specific Extensions

A CCAP is expected to implement vendor-proprietary configuration objects beyond those defined in the present document. 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 (as described in Annex B) in conjunction with a vendor-defined schema. Vendor extensions can also be performed in YANG. A CCAP that supports vendor extension in YANG shall support 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. Refer to Annex K for details on converting a YANG module to XSD.

Modifications to standard configuration objects are allowed within the specific rules defined in Annex B.

See Annex B for requirements related to implementing vendor-specific extensions to the CCAP configuration. Annex B also specifies rules for modifications to standard configuration objects.

6.5.3 CCAP Data Type Definitions

The following data types have been created to support the CCAP configuration object model. See Annex D for the primitive and derived data types used in this model.

Table 6.2: Data Types

Data Type Name	Base Type	Type Constraints	Reference	YANG Data Type
AdminState	Enum	other(1), up(2), down(3), testing(4)	[11]	admin-state-type
AttributeMask	EnumBits	bonded(0), lowLatency(1), highAvailability(2)	[7]	attribute-mask-type
HePidValue	UnsignedShort	(0..8191 | 65535)	[30]	
Host	Choice of IpAddress or InetDomainName			host
IpAddress	InetAddress		[i.12]	inet:ip-address
InetAddressPrefixLength	UnsignedByte	32 or 128 depending on IP version	[i.12]	address-prefix-len-type

Data Type Name	Base Type	Type Constraints	Reference	YANG Data Type
InetIpPrefix	Union of InetIpv4Prefix and InetIpv6Prefix		[i.20]	inet:ip-prefix
InetIpv4Prefix	Union of InetAddressIpv4 and InetAddressPrefixLength	InetAddressPrefixLength : 32	[i.12]	ipv4-prefix
InetIpv6Prefix	Union of InetAddressIpv6 and InetAddressPrefixLength	InetAddressPrefixLength : 128	[i.12]	ipv6-prefix
InetHost	Union of InetIpAddress and InetDomainName		[i.20]	inet:host
InetPortNum	Short		[i.12]	inet:port-number
IPHostPrefix	Union of IPv4HostPrefix and Ipv6HostPrefix			ip-host-prefix
Ipv4HostPrefix				ipv4-host-prefix
Ipv6HostPrefix			[i.16]	ipv6-host-prefix
TenthdB	Short		[20]	
UpDownTrapEnabled	Boolean		IF-MIB in [6]	up-down-trap-enabled

6.5.3.1 AdminState

This data type defines the Admin state RFC 4546 [i.20]. The value of other(1) is used when a vendor extension has been implemented for this attribute.

Reference: RFC 2863 [11].

6.5.3.2 AttributeMask

This data type consists of a sequence of 32-bit positions used to select the bonding group or the channel to which a service flow is assigned. DOCSIS defines three types of Attribute Masks for which this type applies: the Provisioned Attribute Mask that is configured to a Bonding Group or a single-channel, whereas the Required Attribute and the Forbidden Attribute Mask are part of the Service Flow QOS Parameter Set to be matched with the Provisioned Attribute Mask of CMTS-configured Bonding Groups or single-channels. DOCSIS reserves the assignment of the meaning of the first 8 bit positions (left to right) as follows:

- Bit 0: bonded.
- Bit 1: lowLatency.
- Bit 2: highAvailability.
- Bit positions 3-15 are reserved.
- Bit positions 16-31 are freely assigned by operators to represent their own constraints on the channel(s) selection for a particular service flow.

Reference: [7], AttributeMask section.

6.5.3.3 HePidValue

This data type represents a packet identifier (PID) value which ranges from 0 to ($2^{13 - 1}$). The value of 65535 indicates that either the PID is invalid or does not exist.

Reference: ANSI SCTE 154-5 [30].

6.5.3.4 Host

The Host type represents either a strongly-typed IP address or a DNS domain name. Use of this type avoids the weak validation inherent in the union-based inet:host type, as with this type an ip-address cannot be inappropriately validated as a domain-name accidentally. For a particular use of this data type, the CCAP may support only one of these choices: either an IP address or an FQDN.

For attributes with the Host data type, the CCAP shall support configuring an IP address. For attributes with the Host data type, the CCAP should support configuring an FQDN.

6.5.3.5 IpAddress

The IpAddress data type refers to the InetAddress textual convention defined in RFC 4001 [i.12]. It is an octet string of length 4 for an IPv4 address and of length 16 for an IPv6 address. An object of type InetAddress is always interpreted in the context of an object of InetAddressType that selects whether the InetAddress is IPv4 or IPv6.

Reference: RFC 4001 [i.12].

6.5.3.6 InetAddressPrefixLength

This data type corresponds to the InetAddressPrefixLength textual description defined in RFC 4001 [i.12]. It is the number of contiguous "1" bits from the most significant bit of an InetAddress.

Reference: RFC 4001 [i.12].

6.5.3.7 InetIpPrefix

This data type is a union of InetIpv4Prefix and InetIpv6prefix and represents an IP prefix. It is IP version neutral. The format of the textual representations implies the IP version.

6.5.3.8 InetIpv4Prefix

This data type is a union of the InetAddressIpv4 and InetAddressPrefixLength textural conventions defined in RFC 4001 [i.12]. It corresponds to the ipv4-prefix data type defined in RFC 4546 [i.20].

Reference: RFC 4001 [i.12]

6.5.3.9 InetIpv6Prefix

This data type is a union of the InetAddressIpv6 and InetAddressPrefixLength textural conventions defined in RFC 4001 [i.12]. It corresponds to the ipv6-prefix data type defined in RFC 4546 [i.20].

Reference: RFC 4001 [i.12], RFC 4546 [i.20].

6.5.3.10 InetPortNum

This integer represents the port number configured RFC 4001 [i.12]

Reference: RFC 4001 [i.12].

6.5.3.11 InetHost

This data type represents a FQDN, or IPv4 address or IPv6 address and a port number.

Reference: RFC 4546 [i.20]

6.5.3.12 IPHostPrefix

This data type represents an IP host address plus prefix and is IP version neutral. The format of the textual representations implies the IP version. This type is similar to inet:ip-prefix.

This data type is the union of the Ipv4HostPrefix data type and the Ipv6HostPrefix data type.

6.5.3.13 Ipv4HostPrefix

This data type represents an IPv4 host address plus the prefix length, separated by a slash. The prefix length is given by the number following the slash character and is less than or equal to 32. A prefix length valu RFC 4546 [i.20] e of n corresponds to an IP address mask that has n contiguous 1-bits from the most significant bit (MSB) and all other bits set to 0.

This type is derived from the inet:ipv4-prefix type, which has all bits of the IPv4 address set to zero that are not part of the IPv4 prefix. Use of that type requires separate configuration of the interface host address.

The pattern for this looks like: (([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.){3}([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])/(([0-9])|([1-2][0-9])|(3[0-2]))

6.5.3.14 Ipv6HostPrefix

This data type is derived from the inet:ipv6-prefix type, which has all bits of the IPv6 address set to zero that are not part of the IPv6 prefix. Use of that type requires separate configuration of the interface host address. The IPv6 address is represented in the compressed format described in RFC 4291 [i.16], Section 2.2, item 2 with the following additional rules: the "::" substitution is applied to the longest sequence of all-zero 16-bit chunks in an IPv6 address. If there is a tie, the first sequence of all-zero 16-bit chunks is replaced by "::". Single, all-zero 16-bit chunks are not compressed. The canonical format using lowercase characters and leading zeros are not allowed.

Reference: RFC 4291 [i.16].

The pattern for this looks like this:

((:|[0-9a-fA-F]{0,4}):)([0-9a-fA-F]{0,4}:){0,5}' + '((([0-9a-fA-F]{0,4}:)?(:|[0-9a-fA-F]{0,4}))|' + '(((25[0-5]|2[0-4][0-9]|[01]?[0-9]?[0-9])\.){3}' + '(25[0-5]|2[0-4][0-9]|[01]?[0-9]?[0-9])))' + '(/(([0-9])|([0-9]{2})|(1[0-1][0-9])|(12[0-8])))';

pattern '(([^:]+:){6}(([^:]+:[^:]+)|(.*\..*)))|' + '((([^:]+:)*[^:]+)?::(([^:]+:)*[^:]+)?)' + '(/.+)';

6.5.3.15 TenthdB

This data type represents power levels that are normally expressed in dB. Units are in tenths of a dB; for example, 5.1 dB will be represented as 51.

Reference: RFC 4546 [20].

6.5.3.16 UpDownTrapEnabled

Indicates whether linkUp/linkDown traps should be generated for this interface. This is a boolean type, where true means that the trap is enabled.

Reference: RFC 2863 [11], ifLinkUpDownTrapEnable.

6.5.4 CCAP Configuration Objects

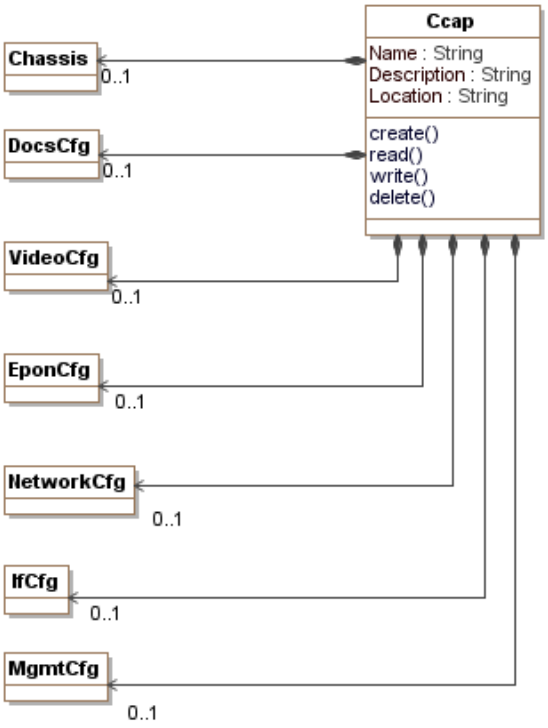


Figure 6.3: CCAP Configuration Objects

6.5.4.1 Ccap Object

The Ccap object serves as the root of the CCAP configuration data. It consists of three attributes that together describe the CCAP platform.

Table 6.3: Ccap Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Name	String	Yes	1..32		
Description	String	Yes			
Location	String	Yes	1..128		

Table 6.4: Ccap Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
Chassis	Directed composition to Chassis		0..1	
DocsCfg	Directed composition to DocsCfg		0..1	
VideoCfg	Directed composition to VideoCfg		0..1	
EponCfg	Directed composition to EponCfg		0..1	
NetworkCfg	Directed composition to NetworkCfg		0..1	
IfCfg	Directed composition to IfCfg		0..1	
MgmtCfg	Directed composition to MgmtCfg		0..1	

6.5.4.1.1 Ccap Object Attributes

6.5.4.1.1.1 Name

This attribute defines the name of the CCAP platform being configured.

6.5.4.1.1.2 Description

This attribute contains the description of the CCAP platform.

6.5.4.1.1.3 Location

This attribute contains any location information for the CCAP.

6.5.4.2 Chassis

This configuration object is included in figure 6.3 for reference. It is defined in clause 6.5.5.2.

6.5.4.3 DocsCfg

This configuration object is included in figure 6.3 for reference. It is defined in clause 6.5.7.1.2.

6.5.4.4 VideoCfg

This configuration object is included in figure 6.3 for reference. It is defined in clause 6.5.6.2.

6.5.4.5 EponCfg

This configuration object is included in figure 6.3 for reference. It is defined in clause 6.5.11.2.

6.5.4.6 NetworkCfg

This configuration object is included in figure 6.3 for reference. It is defined in clause 6.5.8.2.

6.5.4.7 IfCfg

This configuration object is included in figure 6.3 for reference. It is defined in clause 6.5.9.2.

6.5.4.8 MgmtCfg

This configuration object is included in figure 6.3 for reference. It is defined in clause 6.5.10.2.

6.5.5 CCAP Chassis Objects

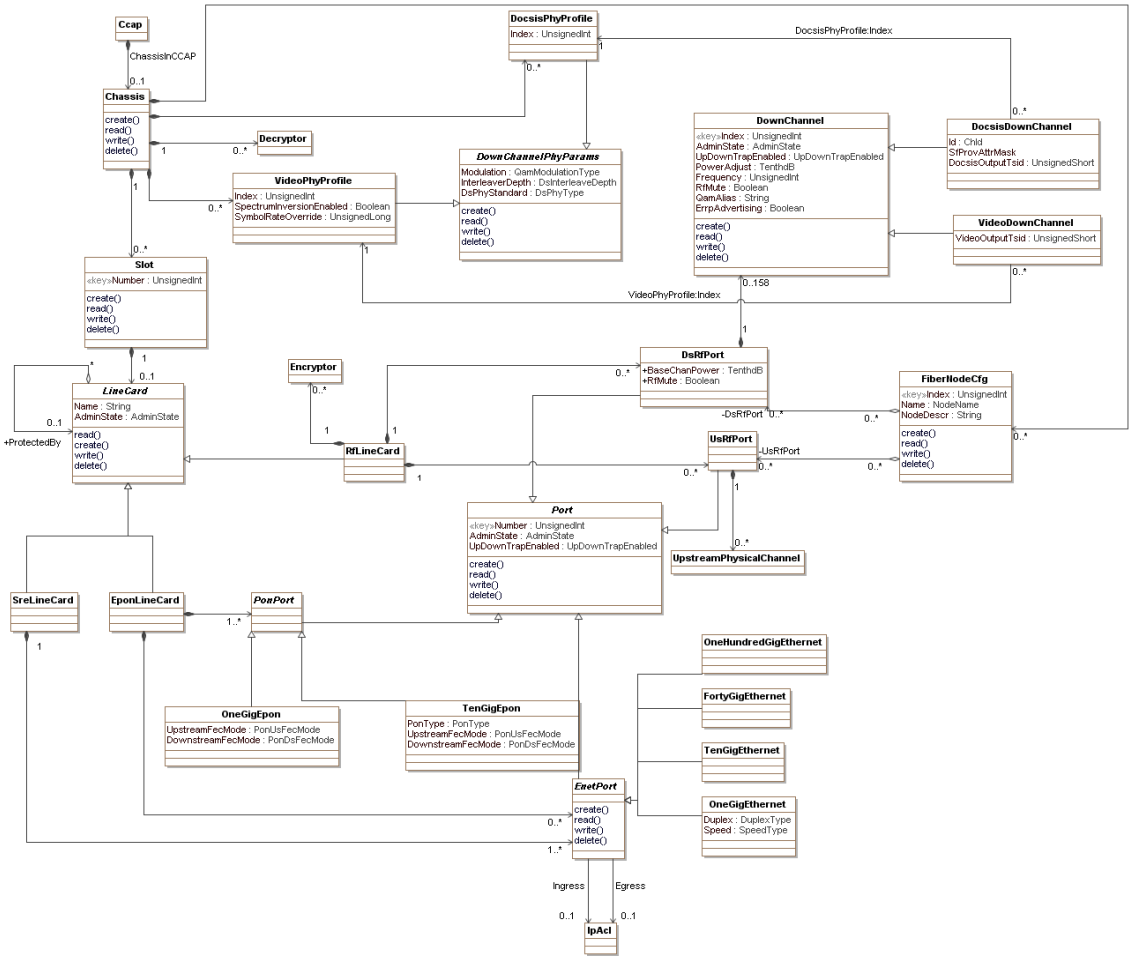


Figure 6.4: CCAP Chassis Objects

6.5.5.1 Ccap

This configuration object is included in figure 6.4 for reference. It is defined in clause 6.5.4.1.

6.5.5.2 Chassis

The Chassis object allows the user to configure the CCAP hardware elements. The Chassis object has the following associations.

Table 6.5: Chassis Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
Slot	Directed composition to Slot	1	0..*	
Decryptor	Directed composition to Decryptor	1	0..*	
FiberNodeCfg	Directed composition to FiberNodeCfg		0..*	
DocsisPhyProfile	Directed composition to DocsisPhyProfile		0..*	
VideoPhyProfile	Directed composition to VideoPhyProfile		0..*	

6.5.5.3 Decryptor

This configuration object is included in figure 6.4 for reference. It is defined in clause 6.5.6.27.

6.5.5.4 Slot

This object configures a slot within the CCAP chassis. Line cards reside in slots.

Table 6.6: Slot Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Number	UnsignedInt	Yes (Key)	0..*		

Table 6.7: Slot Object Associations

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

6.5.5.4.1 Slot Object Attributes

6.5.5.4.1.1 Number

This attribute configures the slot number for which a LineCard object will be configured. The Number attribute is a zero- or one-based index that sequentially numbers the physical slots in the chassis. For example, the Slot numbers start at zero and increase to n-1, where n is the number of slots the chassis supports.

6.5.5.5 LineCard

The abstract object LineCard allows the user to define the common configuration elements for a CCAP line card. There are several types of line cards defined for the CCAP: Downstream Line Card (DLC), Upstream Line Card (ULC), System Route Engine (SRE), a combined Upstream and Downstream line card, and an EPON line card.

Table 6.8: LineCard Abstract Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Name	String	Yes			
AdminState	AdminState	No			down

Line card redundancy or sparing is achieved with a protect relationship between two line cards.

Table 6.9: LineCard Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
LineCard	Directed aggregation to LineCard	*	0..1	ProtectedBy

6.5.5.5.1 LineCard Object Attributes

6.5.5.5.1.1 Name

This attribute stores the name of the line card being configured.

6.5.5.5.1.2 AdminState

This attribute sets the administrative state of the card.

6.5.5.6 RfLineCard

This object holds the configuration data for a specific RF line card, either a downstream line card (DLC), an upstream line card (ULC), or a combined downstream/upstream line card. This object inherits all of the attributes of the LineCard abstract class. A Slot object contains one LineCard object associated with zero or one RfLineCard. A downstream RfLineCard contains one or more DsRfPort; an upstream contains one or more UsRfPort objects; an upstream/downstream RfLineCard contains both DsRfPorts and UsRfPorts. There are several associations for the RfLineCard.

Table 6.10: RfLineCard Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
LineCard	Specialization of LineCard			
Encryptor	Directed composition to Encryptor	1	0..*	
DsRfPort	Directed composition to DsRfPort	1	0..*	
UsRfPort	Directed composition to UsRfPort	1	0..*	
StaticUdpMapEncryption	Directed aggregation to StaticUdpMapEncryption	1	0..*	EnableEncryptionIndex

There are no specific attributes other than what is inherited from the above associations. A minimum lower frequency may be added in a future revision of the present document.

6.5.5.7 EponLineCard

This object configures an EPON line card.

Table 6.11: EponLineCard Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
LineCard	Specialization of LineCard			
PonPort	Directed composition to PonPort		1..*	
EnetPort	Directed composition to EnetPort		0..*	

6.5.5.8 SreLineCard

The SRE line card is the name given to the line card in the CCAP chassis that contains all the NSI and Management functions for the CCAP. This line card is associated with at least one EnetPort, which serves as the NSI. This object inherits all of the attributes of the LineCard abstract object. There are two associations for the SRE.

Table 6.12: SreLineCard Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
LineCard	Specialization of LineCard			
EnetPort	Directed composition to EnetPort	1	1..*	

6.5.5.9 Encryptor

This configuration object is included in figure 6.4 for reference. It is defined in clause 6.5.6.34.

6.5.5.10 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.

Table 6.13: Port Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Number	UnsignedInt	Yes (Key)	0..*		
AdminState	AdminState	No			down
UpDownTrapEnabled	UpDownTrapEnabled	No			false

6.5.5.10.1 Port Object Attributes

6.5.5.10.1.1 Number

The Number attribute of Port is a zero- or one-based index that sequentially numbers the physical ports of each derived type. For example, the port numbers of the DsRfPort objects start at zero and increase to n-1, where n is the total number of DsRfPorts.

6.5.5.10.1.2 AdminState

This attribute configures the administrative state of the physical port.

6.5.5.10.1.3 UpDownTrapEnabled

This attribute configures whether linkUp/linkDown traps are enabled for this port.

6.5.5.11 DsRfPort

This object allows for the configuration of a physical Downstream RF port on an RfLineCard. The DsRfPort is a type of the abstract class Port and inherits those common parameters. In the CCAP, a single port now encompasses the entire downstream spectrum instead of a few carriers as are seen in the current generation EQAM and CMTS products. A DsRfPort object contains the attributes in the following table.

Table 6.14: DsRfPort Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
BaseChanPower	TenthdB	No		TenthdB	vendor-specific
RfMute	Boolean	No			false

Table 6.15: DsRfPort Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
Port	Specialization of Port			
DownChannel	Directed composition to DownChannel	1	0..158	

6.5.5.11.1 DsRfPort Object Attributes

6.5.5.11.1.1 BaseChanPower

This attribute configures the base output power for each single DownChannel on the DsRfPort. The value is expressed in dBmV in units of TenthdB. The default value is vendor-specific. Acceptable power ranges for this attribute are defined in EN 302 878-3 [i.2], clause 6.3.5.1.1, Power per Channel CMTS or EQAM.

Reference: EN 302 878-3 [i.2], Power per Channel CMTS or EQAM section

6.5.5.11.1.2 RfMute

The attribute RfMute refers to a diagnostic state defined in the DOCSIS RF Interface EN 302 878-3 [i.2] specification. Muting an RF port affects only the output power and does not impact the operational status of any channel on the port.

6.5.5.12 DownChannel

The DownChannel object contains the attributes used when configuring a QAM channel. This object is contained within a DsRfPort.

Table 6.16: DownChannel Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
ChannelIndex	UnsignedInt	Yes (Key)	0..158		
AdminState	AdminState	No			down
UpDownTrapEnabled	UpDownTrapEnabled	No			true
PowerAdjust	TenthdB	No		TenthdB	0
Frequency	UnsignedInt	Yes		Hertz	
RfMute	Boolean	No			false
QamAlias	String	No			""
ErrpAdvertising	Boolean	No			true

A DsRfPort contains a number of configured DownChannel objects. A DownChannel is either a VideoDownChannel or a DosisDownChannel. The PHY parameters for a down channel are specified by associating a down channel with a PHY profile, either a VideoPhyProfile or DocsisPhyProfile, depending on the down channel type. If a PHY profile is not specified, the CCAP will provide vendor-specific PHY defaults. A DownChannel is a generalization of either a VideoDownChannel or a DocsisDownChannel.

Table 6.17: DownChannel Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
ErmParams	Directed composition to ErmParams		0..l	

6.5.5.12.1 DownChannel Object Attributes

6.5.5.12.1.1 ChannelIndex

This key identifies a downstream channel on a specific downstream RF Port.

6.5.5.12.1.2 AdminState

This attribute represents the administrative status of the channel. Setting the value to down(3) results in the channel being muted. The value of testing(4) is used to generate a continuous test wave on this QAM channel.

6.5.5.12.1.3 UpDownTrapEnabled

This attribute configures whether linkUp/linkDown traps are enabled for this channel.

6.5.5.12.1.4 PowerAdjust

This attribute represents the power gain for the channel. It is expressed in TenthdB.

6.5.5.12.1.5 Frequency

This attribute specifies the center frequency of the channel. It is expressed in Hertz. The CCAP shall reject the configuration of a DownChannel instance that overlaps in frequency with another DownChannel instance on the same downstream RF port.

6.5.5.12.1.6 RfMute

This attribute configures the mute state for the specific DownChannel. If set to true, the ifOperStatus of the VideoDownChannel or DocsisDownChannel associated with this instance of DownChannel is set to "down". If set to false, no muting takes place. Operation while muted is described in EN 302 878-3 [i.2].

6.5.5.12.1.7 QamAlias

This attribute represents the name of the QAM channel and is equivalent to the ifAlias object in the if-MIB.

6.5.5.12.1.8 ErrpAdvertising

This attribute represents the Edge Resource Resgistration Period (ERRP) advertisement state of the QAM channel. If set to true, the QAM channel is advertised; otherwise it is not advertised. This is primarily useful when statically configuring the QAM channels and when the QAM channel is not made part of the ERM channel list. This attribute is optional for DocsisDownChannel.

6.5.5.12.2 DownChannel Configuration Constraints

The CCAP shall reject activation of a set of configuration objects that would attempt to enable more than one QAM channel with the same center frequency on any single downstream RF port.

There are two types of QAM in the CCAP device regarding the advertisement to the ERM.

- pilot QAM that are advertised;
- replicated QAM that are not advertised.

In the CCAP configuration model, there are two types of Output TSIDs: VideoOutputTsid, required for all VideoDownChannel instances, and DocsisOutputTsid, an optional attribute of a DocsisDownChannel. The CCAP may reject configurations that cause the same Output TSID value to be advertised to the same ERM more than once; therefore, exactly one pilot QAM is advertised to the ERM per replication group. If the CCAP allows configurations in which the same output TSID is configured to be advertised to the ERM for multiple down channels, then the CCAP shall only advertise one of those TSIDs to the ERM as a pilot QAM. The CCAP will use vendor-proprietary rules to decide which QAM to advertise as the pilot in this case.

When a change in configuration results in a replicated QAM transitioning to a pilot QAM, the CCAP shall advertise the transitioned QAM as a new resource to the ERM.

When a change in configuration results in a pilot QAM transitioning to a replicated QAM, the CCAP shall notify the ERM and delete the corresponding QAM resource from the ERM. This notification takes place so the sessions can be properly torn down and repositioned.

When advertising the pilot QAM to the ERM, the CCAP shall include a list all fiber nodes to which it is replicated.

Output TSIDs are unique per DsRfPort. Therefore, when the CCAP replicates a QAM, the CCAP shall de-advertise that QAM from the ERM.

The CCAP shall reject configurations of Output TSIDs values that are not unique on a specific DsRfPort.

The CCAP shall support the configuration of whether or not duplicate Output TSID values are allowed on the CCAP.

6.5.5.12.3 Output Replication Requirements

An input transport stream is a sequence of MPEG frames received at a single IP address and UDP port by the CCAP. An input transport stream typically consists of a set of programs that are each identified by an input program number. Each input program consists of a number of elementary streams, each individually identified by a PID. An input transport stream may contain elementary streams that are not part of a program.

A VideoInputTs object configures an input transport stream. A UnicastVideoInputTs object configures a unicast input transport stream; a MulticastVideoInputTs object configures a multicast input transport stream.

An output transport stream is defined as a sequence of MPEG frames transmitted by a CCAP. An output transport stream typically consists of multiple output programs. Each output program consists of a set of elementary streams each identified by an individual PID. An output Multi-Program Transport Stream (MPTS) is an output transport stream that contains tables that identify its programs and associated elementary streams. An output TSID is a 16-bit number that uniquely identifies a MPTS in a streaming zone.

A VideoOutputTs object statically configures a video output transport stream on the CCAP. A VideoOuputTs object is identified with a CCAP-unique Index. A VideoOutputTs object is statically associated with either MptsPassThruSession instances or can be configured as an MPTS that multiplexes several ProgramSession instances and/or PidSession instances. VideoOutputTs instances are only associated with sessions, not directly with video input transport streams. A VideoOutputTs instance is associated with a VideoDownChannel instance, configured with a VideoOutputTsid that is included in its PAT, as transmitted by the CCAP.

A ProgramSession object statically configures the mapping of input transport streams to one or more VideoOutputTs instances. A PidSession object statically configures the mapping of input elementary streams to VideoOutputTs instances. An MptsPassThruSession object statically configures the mapping of an entire input MPTS to VideoOutputTs instances.

It is expected that a given MPTS identified by a unique VideoOutputTs Index can be replicated on more than one CCAP RF port. For example, a narrowcast VOD or SDV MPTS may be transmitted to two, three, or four CCAP downstream RF ports, while digital broadcast video content may be replicated to most or all CCAP downstream RF ports.

A VideoOutputTs instance is statically configured to one or more VideoDownChannel instances via its association to the VideoDownChannel instances in which it will be included. Each VideoDownChannel object represents the contents transmitted on a single RF port at a single frequency. The CCAP shall replicate the output transport stream represented by a VideoOutputTs object to all of the QAM channels represented by the VideoDownChannel objects to which the VideoOutputTs is associated.

Depending on CCAP vendor implementation, the CCAP may transmit the replicated MPEG packets of the multiplexed set of video sessions in exactly the same order.

The CCAP shall meet all MPEG requirements, per ISO/IEC 13818-1 [5], for replicated video sessions.

The CCAP should allow the configuration of different frequencies and DownChannelPhyParams for different VideoDownChannels to which a VideoOutputTs instance is associated.

The CCAP may reject a configuration in which a VideoOutputTs is associated with VideoDownChannel instances that reside on different frequencies.

6.5.5.13 DocsisDownChannel

The DocsisDownChannel object is a DownChannel used exclusively for DOCSIS. The DownChannel is its generalization.

The DocsisDownChannel object is a specialization of DownChannel.

Some CCAP implementations may implement the association of non-primary capable downstream channels with MAC domain indirectly, based on RF plant topology configuration. In such a case CCAP device may ignore configuration settings communicated through the label Non-PrimaryCapableDs. If a DocsisDownChannel is not associated with a DocsisPhyProfile instance, the CCAP provides vendor-specific PHY defaults.

Table 6.18: DocsisDownChannel Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Id	UnsignedInt	Yes	0..255		
SfProvAttrMask	AttributeMask	Yes			
DocsisOutputTsid	UnsignedShort	No			0

Table 6.19: DocsisDownChannel Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
DownChannel	Specialization of DownChannel			
DocsisPhyProfile	Directed composition to DocsisPhyProfile	0..*	1	DocsisPhyProfileIndex

6.5.5.13.1 DocsisDownChannel Object Attributes

6.5.5.13.1.1 Id

Unique identifier for the DocsisDownChannel. A value of 0 (zero) means that the CCAP will automatically assign the Id.

6.5.5.13.1.2 SfProvAttrMask

This attribute contains Provisioned Attribute Mask of non-bonded service flow assignment to this channel.

6.5.5.13.1.3 DocsisOutputTsid

This attribute specifies the optional output TSID of the channel. The TSID is globally unique per CCAP. Replicated output streams share the same Output TSID.

6.5.5.14 VideoDownChannel

The VideoDownChannel object is a DownChannel used exclusively for video channel configuration. If a VideoDownChannel is not associated with an instance of VideoPhyProfile, the CCAP provides vendor-specific defaults.

Table 6.20: VideoDownChannel Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
VideoOutputTsid	UnsignedShort	Yes			

Table 6.21: VideoDownChannel Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
DownChannel	Specialization of DownChannel			
VideoPhyProfile	Directed composition to VideoPhyProfile	0..*	1	VideoPhyProfileIndex

6.5.5.14.1 VideoDownChannel Object Attributes

6.5.5.14.1.1 VideoOutputTsid

This attribute specifies the output TSID of the channel and is required for a VideoDownChannel. The TSID is globally unique per CCAP. Replicated output streams share the same Output TSID.

6.5.5.15 DocsisPhyProfile

The DocsisPhyProfile object is a specialization of the DownChannelPhyParams object and allows PHY parameters to be specified for a DocsisDownChannel instance.

Table 6.22: DocsisPhyProfile Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes			

Table 6.23: DocsisPhyProfile Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
DownChannelPhyParams	Specialization of DownChannelPhyParams			

6.5.5.15.1 DocsisPhyProfile Object Attributes

6.5.5.15.1.1 Index

This attributes specifies a unique index for this instance of DocsisPhyProfile.

6.5.5.16 VideoPhyProfile

The VideoPhyProfile object is a specialization of the DownChannelPhyParams object and allows PHY parameters to be specified for a VideoDownChannel instance.

Table 6.24: VideoPhyProfile Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes			
SpectrumInversion	Boolean	No			false
SymbolRateOverride	UnsignedLong	No		Symbols per second	

Table 6.25: VideoPhyProfile Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Units	Default Value
DownChannelPhyParams	Specialization of DownChannelPhyParams				

6.5.5.16.1 VideoPhyProfile Object Attributes

6.5.5.16.1.1 Index

This attributes specifies a unique index for this instance of VideoPhyProfile.

6.5.5.16.1.2 SpectrumInversion

This attribute specifies RF Signal Spectrum inversion. When set to true, it indicates that the QAM channel spectrum is inverted.

6.5.5.16.1.3 SymbolRateOverride

This attribute allows the default symbol rate for the VideoPhyProfile to be overridden, expressed in symbols per second. If not specified, channels configured to use this VideoPhyProfile operate with the value specified by DOCSIS for the Annex and modulation.

6.5.5.17 DownChannelPhyParams

DownChannelPhyParams is an abstract object that can be used to specify the physical attributes of a DownChannel.

Table 6.26: DownChannelPhyParams Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Modulation	Enum	No	other(1), qam64(2), qam128(3), qam256(4), qam512(5), qam1024(6)		qam256
InterleaverDepth	Enum	No	other(1), fecI8J16(2), fecI12J17(3), fecI16J8(4), fecI32J4(5), fecI64J2(6), fecI128J1(7), fecI128J2(8), fecI128J3(9), fecI128J4(10), fecI128J5(11), fecI128J6(12), fecI128J7(13), fecI128J8(14)		fecI128J1
DsPhyStandard	Enum	No	other (1), dvbc(2), j83annexB(3), j83annexC(4)		j83annexB

6.5.5.17.1 DownChannelPhyParams Object Attributes

6.5.5.17.1.1 Modulation

Defines the modulation type used. The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.5.17.1.2 InterleaverDepth

This attribute represents the interleaving depth or operation mode of the interleaver. The value of other(1) is used when a vendor-extension has been implemented for this attribute.

This attribute is ignored when DsPhyStandard has a value other than j83annexB(3).

6.5.5.17.1.3 DsPhyStandard

This attribute specifies the standard supported by the QAM channel. A value of dvbc(2) corresponds to J.83 Annex A. The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.5.18 FiberNodeCfg

The FiberNodeCfg Object is defined in [7] and is used in CCAP with one change to include an index for an instance of a fiber node, defined here.

It defines the cable hybrid fiber/coax system (HFC) plant Fiber Nodes reached by RF ports on a CCAP.

This object supports the creation and deletion of multiple instances.

Reference: [7], FiberNodeCfg Object section.

Table 6.27: FiberNodeCfg Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes (Key)			

The FiberNodeCfg object has the following associations.

Table 6.28: FiberNodeCfg Object Associations

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

6.5.5.18.1 FiberNodeCfg Object Attributes

6.5.5.18.1.1 Index

The index of the fiber node being configured.

6.5.5.19 UsRfPort

A UsRfPort object represents a physical upstream RF connector on an RfLineCard. It is derived from the Port abstract class, and so inherits all attributes of that class, including its associations. A UsRfPort is contained by an RfLineCard. It contains one or more UpstreamPhysicalChannel objects. This object has no attributes other than what has been inherited from the abstract class *Port*, but does have several associations.

Table 6.29: UsRfPort Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
Port	Specialization of Port			
UpstreamPhysicalChannel	Directed composition to UpstreamPhysicalChannel	1	0..*	

6.5.5.20 UpstreamPhysicalChannel

This configuration object is included in figure 6.4 for reference. It is defined in clause 6.5.7.8.6.

6.5.5.21 EnetPort

The EnetPort object is an abstract class that allows an Ethernet port to be configured on a line card that contains Ethernet ports. EnetPort is also a type of the abstract class Port. Ethernet ports are associated with the SreLineCard and the EponLineCard.

Table 6.30: EnetPort Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
Port	Specialization of Port			
IpInterface	Directed composition to IpInterface		0..1	
IpAcl	Directed association to IpAcl		0..1	Ingress
IpAcl	Directed association to IpAcl		0..1	Egress

6.5.5.22 OneGigEthernet

This object configures a one gigabit interface for an Ethernet port. The speed and duplex settings for this type of port can be configured via this object.

Table 6.31: OneGigEthernet Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Duplex	Enum	No	other(1), fullDuplex(2), halfDuplex(3), autoNegotiated(4)		fullDuplex
Speed	Enum	Yes	other(1), tenMbitEthernet(2), hundredMbitEthernet(3), oneGigabit(4), auto(5)		

Table 6.32: OneGigEthernet Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
EnetPort	Specialization of EnetPort			

6.5.5.22.1 OneGigEthernet Object Attributes

6.5.5.22.1.1 Duplex

This attribute configures the Ethernet DuplexState of the interface. The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.5.22.1.2 Speed

This attribute configures the speed of the interface for interfaces that can support multiple speeds. The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.5.23 TenGigEthernet

This object configures a ten gigabit interface for an Ethernet port.

Table 6.33: TenGigEthernet Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
EnetPort	Specialization of EnetPort			

6.5.5.24 FortyGigEthernet

This object configures a 40 gigabit interface for an Ethernet port.

Table 6.34: FortyGigEthernet Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
EnetPort	Specialization of EnetPort			

6.5.5.25 OneHundredGigEthernet

This object configures a 100 gigabit interface for an Ethernet port.

Table 6.35: OneHundredGigEthernet Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
EnetPort	Specialization of EnetPort			

6.5.5.26 PonPort

This abstract configuration object allows for an EPON port to be configured on an EPON line card. PonPort is a type of the abstract class Port.

Table 6.36: PonPort Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
Port	Specialization of Port			

6.5.5.27 OneGigEpon

This configuration object allows for a one Gigabit EPON port to be configured on an EPON line card. It is a type of the abstract class PonPort.

Table 6.37: OneGigEpon Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
UpstreamFecMode	Enum	No	other(1), enabled(2), disabled(3), perOnu(4)		disabled
DownstreamFecMode	Enum	No	other(1), enabled(2), disabled(3), perOnu(4)		disabled

Table 6.38: OneGigEpon Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
PonPort	Specialization of PonPort			

6.5.5.27.1 PonPort Object Attributes

6.5.5.27.1.1 UpstreamFecMode

This attribute configures the FEC mode applied to the EPON upstream. The perOnu option allows the ONU provisioning process to determine whether FEC should be enabled or disabled. This option is only valid for 1G EPON interfaces.

The default value for the 1G EPON interface is disabled(3).

The default value for the 10G EPON interface is enabled(2).

The perOnu option allows the ONU provisioning process to determine whether FEC should be enabled or disabled. This option is only valid for 1G EPON interfaces.

The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.5.27.1.2 DownstreamFecMode

This attribute configures the FEC mode of the EPON downstream. The perOnu option allows the ONU provisioning process to determine whether FEC should be enabled or disabled. This option is only valid for 1G EPON interfaces.

The perOnu option allows the ONU provisioning process to determine whether FEC should be enabled or disabled. This option is only valid for 1G EPON interfaces.

The default value for the 1G EPON interface is disabled(3).

The default value for the 10G EPON interface is enabled(2).

The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.5.28 TenGigEpon

This configuration object allows for a symmetric or asymmetric ten Gigabit EPON port to be configured on an EPON line card. It is a type of the abstract class PonPort.

Table 6.39: TenGigEpon Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
PonType	Enum	Yes	other(1), symmetric10x10(2), asymmetric10x1(3)		
UpstreamFecMode	Enum	No	other(1), enabled(2), disabled(3), perOnu(4)		enabled
DownstreamFecMode	Enum	No	other(1), enabled(2), disabled(3), perOnu(4)		enabled

Table 6.40: TenGigEpon Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
PonPort	Specialization of PonPort			

6.5.5.28.1 TenGigEpon Object Attributes

6.5.5.28.1.1 PonType

This attribute configures the speed of the 10G EPON interfaces on the line card and allows for asymmetrical upstream and downstream speeds. The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.5.28.1.2 UpstreamFecMode

This attribute configures the FEC mode applied to the EPON upstream. The perOnu option allows the ONU provisioning process to determine whether FEC should be enabled or disabled. This option is only valid for 1G EPON interfaces.

The default value for the 1G EPON interface is disabled(3).

The default value for the 10G EPON interface is enabled(2).

The perOnu option allows the ONU provisioning process to determine whether FEC should be enabled or disabled. This option is only valid for 1G EPON interfaces.

The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.5.28.1.3 DownstreamFecMode

This attribute configures the FEC mode of the EPON downstream. The perOnu option allows the ONU provisioning process to determine whether FEC should be enabled or disabled. This option is only valid for 1G EPON interfaces.

The perOnu option allows the ONU provisioning process to determine whether FEC should be enabled or disabled. This option is only valid for 1G EPON interfaces.

The default value for the 1G EPON interface is disabled(3).

The default value for the 10G EPON interface is enabled(2).

The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.6 CCAP Video Session Configuration Objects

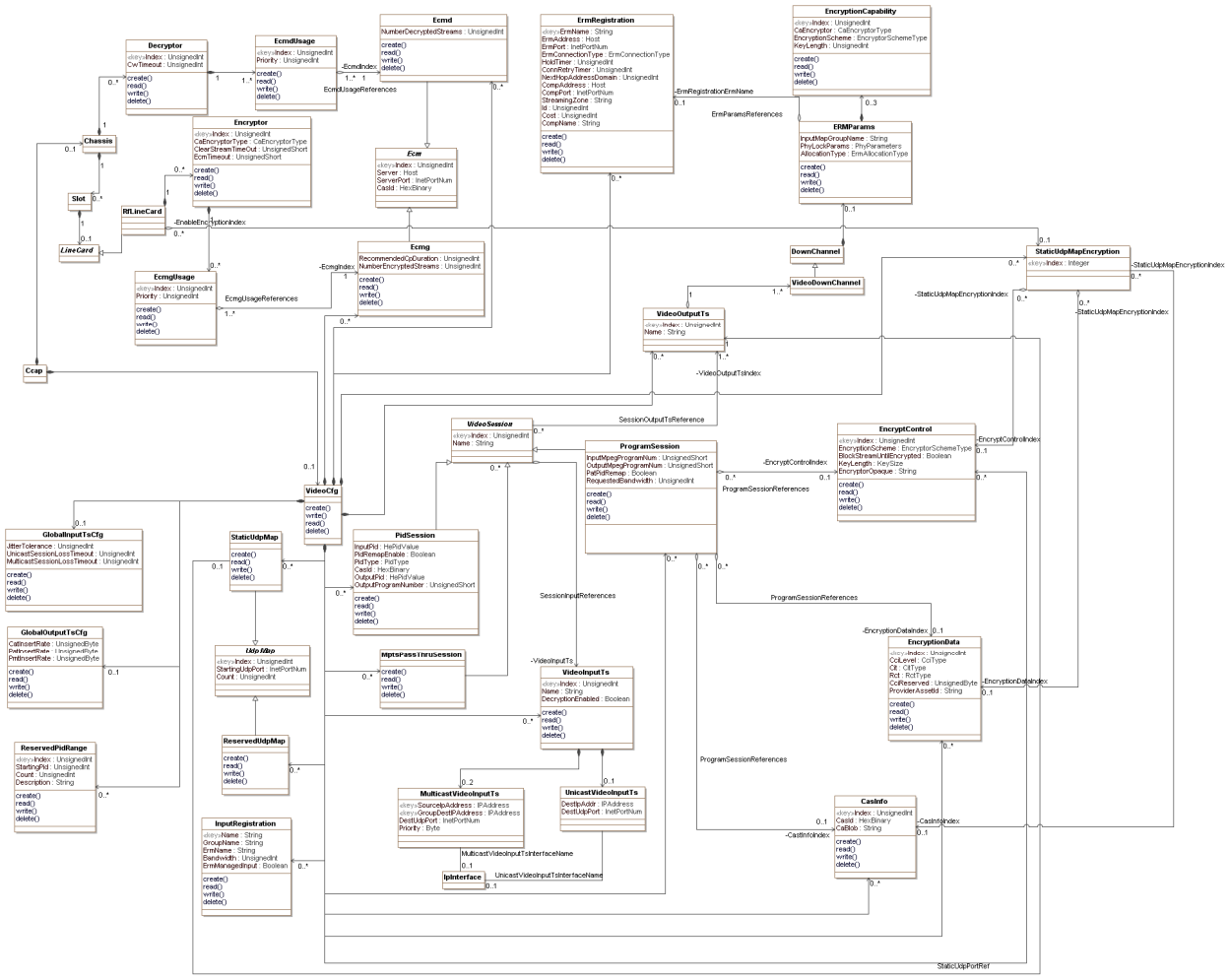


Figure 6.5: CCAP Video Session Configuration Objects

6.5.6.1 Ccap

This configuration object is included in figure 6.5 for reference. It is defined in clause 6.5.4.1.

6.5.6.2 VideoCfg

The VideoCfg object is the primary container of video configuration objects. It has the following associations:

Table 6.41: VideoCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
GlobalInputTsCfg	Directed composition to GlobalInputTsCfg		0..1	
GlobalOutputTsCfg	Directed composition to GlobalOutputTsCfg		0..1	
StaticUdpMap	Directed composition to StaticUdpMap		0..*	
ReservedUdpMap	Directed composition to ReservedUdpMap		0..*	
ReservedPidRange	Directed composition to ReservedPidRange		0..*	
InputRegistration	Directed composition to InputRegistration		0..*	
ProgramSession	Directed composition to ProgramSession		0..*	
MptsPassThruSession	Directed composition to MptsPassThruSession		0..*	
PidSession	Directed composition to PidSession		0..*	
VideoInputTs	Directed composition to VideoInputTs		0..*	
CasInfo	Directed composition to CasInfo		0..*	

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
EncryptionData	Directed composition to EncryptionData		0..*	
EncryptControl	Directed composition to EncryptControl		0..*	
ErmRegistration	Directed composition to ErmRegistration		0..*	
VideoOutputTs	Directed composition to VideoOutputTs		0..*	
Ecmg	Directed composition to Ecmg		0..*	
Ecmd	Directed composition to Ecmd		0..*	
StaticUdpMapEncryption	Directed composition to StaticUdpMapEncryption		0..1	

6.5.6.3 GlobalInputTsCfg

This object represents global configuration of input transport streams.

Table 6.42: GlobalInputTsCfg Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
JitterTolerance	UnsignedInt	No		milliseconds	100
UnicastSessionLossTimeout	UnsignedInt	No		milliseconds	5 000
MulticastSessionLossTimeout	UnsignedInt	No		milliseconds	5 000

6.5.6.3.1 GlobalInputTsCfg Object Attributes

6.5.6.3.1.1 JitterTolerance

This attribute represents the acceptable delay variation in milliseconds for incoming streams. The jitter option sets the size of a dejittering buffer that absorbs the input jitter of a session.

6.5.6.3.1.2 UnicastSessionLossTimeout

This attribute represents the loss of signal timeout in milliseconds for unicast input streams. See ANSI SCTE 154-4 [29], mpegLossOfSignalTimeout.

6.5.6.3.1.3 MulticastSessionLossTimeout

This attribute represents the loss of signal timeout in milliseconds for the multicast input streams.

6.5.6.4 GlobalOutputTsCfg

This object represents global configuration of output transport streams.

Table 6.43: GlobalOutputTsCfg Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
CatInsertRate	UnsignedByte	No	0..32	tables/second	10
PatInsertRate	UnsignedByte	No	0..32	tables/second	10
PmtInsertRate	UnsignedByte	No	0..32	tables/second	10

6.5.6.4.1 GlobalOutputTsCfg Object Attributes

6.5.6.4.1.1 CatInsertRate

This attribute represents the CAT insertion rate expressed in tables/second (see ANSI SCTE 154-4 [29], mpegOutputTSCatInsertRate).

6.5.6.4.1.2 PatInsertRate

This attribute represents the PAT table interval expressed in tables/second (see ANSI SCTE 154-4 [29], mpegOutputTSPatInsertRate).

6.5.6.4.1.3 PmtInsertRate

This attribute represents the PMT table interval expressed in tables/second (see ANSI SCTE 154-4 [29], mpegOutputTSPatInsertRate).

6.5.6.5 UdpMap

This abstract object holds the UDP attributes that are used in the StaticUdpMap and ReservedUdpMap objects.

Table 6.44: UdpMap Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes (Key)			
StartingPort	InetPortNum	No			0
Count	UnsignedInt	No			0

6.5.6.5.1 UdpMap Object Attributes

6.5.6.5.1.1 Index

This key represents a globally-unique identifier of the object instance.

6.5.6.5.1.2 StartingPort

This attribute represents the UDP port range start value.

6.5.6.5.1.3 Count

This attribute represents the number of UDP ports starting from the StartingPort attribute value.

6.5.6.6 StaticUdpMap

This object represents the UDP port ranges used for static video sessions. It is a specialization of UdpMap.

Table 6.45: StaticUdpMap Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
UdpMap	Specialization of UdpMap			
VideoOutputTs	Directed association to VideoOutputTs	0..1	1	StaticUpdPortRef

6.5.6.7 ReservedUdpMap

This object represents reserved ports to be used for non-video applications. It is a specialization of UdpMap.

Table 6.46: ReservedUdpMap Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
UdpMap	Specialization of UdpMap			

6.5.6.8 ReservedPidRange

This object represents reserved PID range to not be used on ERM assignments.

Table 6.47: ReservedPidRange Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes (Key)			
StartingPid	UnsignedInt	No			0
Count	UnsignedInt	No			0
Description	String	No			""

6.5.6.8.1 ReservedPidRange Object Attributes

6.5.6.8.1.1 Index

This key represents the unique identifier of an instance of this object.

6.5.6.8.1.2 StartingPid

This attribute represents the PID range starts for other applications' reserved PIDs.

6.5.6.8.1.3 Count

This attribute represents the number of reserved PIDs starting from the StartingPid attribute value.

6.5.6.8.1.4 Description

This attribute represents the description associated with a PID range configured instance.

6.5.6.9 InputRegistration

This object represents the configuration of Edge ERRP parameters.

Table 6.48: InputRegistration Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Name	String	Yes (Key)			
GroupName	String	No			""
ErmName	String	No			""
Bandwidth	UnsignedInt	No			0
ErmManagedInput	Boolean	Yes			

6.5.6.9.1 InputRegistration Object Attributes

6.5.6.9.1.1 Name

This key represents the Input interface name. This name corresponds to the RFC 4133 [14], ENTITY-MIB entPhysicalName.

6.5.6.9.1.2 GroupName

This attribute represents the name of the Edge Input Group associated with this input. This parameter is used in the ERRP Edge Input attribute.

6.5.6.9.1.3 ErmName

This attribute represents the ERM where the QAM channel is advertised. If empty, the QAM channel is not advertised.

6.5.6.9.1.4 Bandwidth

This attribute represents the bandwidth of the edge input to be advertised. If zero or not present, the CCAP advertises the full bandwidth of the edge input. If the attribute ErmManagedInput is set to false, operators should set this attribute to a value that greatly exceeds the speed of the input interface; this will cause the ERM to not actively manage the input bandwidth.

6.5.6.9.1.5 ErmManagedInput

This attribute allows the Operator to configure whether or not the ERM should manage the input bandwidth on this EdgeInput Interface. A value of true indicates that the ERM will manage the input bandwidth; a value of false indicates that the CCAP will manage the input bandwidth. If set to false, operators should set the Bandwidth attribute to a value that greatly exceeds the speed of the input interface so that the ERM will not actively manage the input bandwidth.

6.5.6.10 CasInfo

The CasInfo object serves two purposes:

1) It identifies the ECMG(s) that need(s) to be involved in the encryption of the program session. In the case of a Simulcrypt operation, more than one CasInfo object can be attached to the same ProgramSession.

2) It defines a CA-specific opaque object that needs to be forwarded to the appropriate ECMG when the session is initialized.

A CasInfo object contains the attributes in table 6.49.

Table 6.49: CasInfo Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes (Key)			
CasId	HexBinary	Yes	size(8)		
CaBlob	String	Yes			

6.5.6.10.1 CasInfo Object Attributes

6.5.6.10.1.1 Index

This attribute configures the index for an instance of CasInfo for a given ProgramSession.

6.5.6.10.1.2 CasId

CasId is the hexadecimal representation of the CAS system identifier.

6.5.6.10.1.3 CaBlob

CaBlob is opaque data that the Encryptor is required to forward to the ECMG associated with the specified CasId.

6.5.6.11 EncryptionData

The EncryptionData object allows a per video session encryption configuration.

Table 6.50: EncryptionData Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes (Key)			
CciLevel	Enum	Yes	other(1), copyFreely(2), copyOneGeneration(3), copyNoMore(4), copyNever(5)		
Cit	Enum	Yes	other(1), clear(2), set(3)		
Rct	Enum	Yes	other(1), notAsserted(2), required(3)		
CciReserved	UnsignedByte	Yes	0..3		
ProviderAssetId	String	Yes	1..255		

6.5.6.11.1 EncryptionData Object Attributes

6.5.6.11.1.1 Index

The index is the key for the EncryptionData object.

6.5.6.11.1.2 CciLevel

This attribute represents the Copy Control Indicator/Digital Rights protection applicable to the program. It is forwarded to all active ECMGs to be encapsulated into ECMs. The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.6.11.1.3 Cit

This attribute represents the Constrained Image Trigger flag applicable to the program. It is forwarded to all active ECMGs to be encapsulated into ECMs. The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.6.11.1.4 Rct

This attribute represents the Redistribution Control Trigger flag applicable to the program. It is forwarded to all active ECMGs to be encapsulated into ECMs. The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.6.11.1.5 CciReserved

This attribute reserves 2 bits of copy control information (CCI) for future use. It is forwarded to all active ECMGs to be encapsulated into ECMs.

6.5.6.11.1.6 ProviderAssetId

This attribute configures the Provide Asset Id parameter that is passed in the initial RTSP session SETUP (e.g. for VOD) to the Encryptor and enables the Encryptor to uniquely identify/reference the VOD asset or broadcast channel.

6.5.6.12 EncryptControl

This configuration object selects the encryption option of a static encryption session.

Table 6.51: EncryptControl Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes (Key)			
EncryptionScheme	Enum	Yes	other(1), des(2), aes(3), 3des(4), dvbcsa(5), dvbcsa3(6)		
BlockStreamUntilEncrypted	Boolean	No			true
KeyLength	Enum	Yes	other(1), 56bits(2), 128bits(3), 192bits(4), 256bits(5)		
EncryptorOpaque	String	Yes			

6.5.6.12.1 EncryptControl Object Attributes

6.5.6.12.1.1 Index

This attribute configures the index for an instance of EncryptControl for a given ProgramSession.

6.5.6.12.1.2 EncryptionScheme

This attribute defines the encryption algorithm to be used for a given video session. The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.6.12.1.3 BlockStreamUntilEncrypted

BlockStreamUntilEncrypted indicates if the encryption engine should block the program until it can encrypt it (i.e. it has received a first Entitlement Control Message (ECM) and Control Word (CW) from the ECMG) or release it in the clear to the destination or output. Values are 0 meaning false or 1 meaning true. Note that this parameter can be used to enforce synchronous behavior, wherein the RTSP server (i.e. Encryption Engine) will not acknowledge the session request back to the ERM until it has actually started to encrypt the stream. Obviously, this assurance comes at the expense of setup latency.

KeyLength

This attribute configures the number of bits in the encryption keys used by encryption algorithm defined by the EncryptionScheme attribute. The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.6.12.1.4 EncryptorOpaque

EncryptorOpaque holds private data used by the Encryptor that may be under CA license from the CA vendor.

6.5.6.13 VideoInputTs

The VideoInputTs object configures a given MPEG-2 Transport stream that may be unicast or multicast. Each VideoInputTs object shall have either:

- one or two MulticastVideoInputTs objects associated with it;
- one UnicastVideoInputTs object associated with it.

Having two MulticastVideoInputTs objects associated with it occurs when input TS redundancy is configured (Hot-Hot sparing).

Table 6.52: VideoInputTs Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes (Key)			
Name	String	No			""
DecryptionEnabled	Boolean	No			false

When redundancy of the input multicast TS is configured, a VideoInputTs object is associated with two MulticastVideoInputTs objects. A VideoInputTs object can also be referenced from multiple ProgramSession, MptsPassThruSession, or PidSession objects.

Table 6.53: VideoInputTs Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
MuticastVideoInputTs	Directed composition to MulticastVideoInputTs		0..2	
UnicastVideoInputTs	Directed composition to UnicastVideoInputTs		0..1	

6.5.6.13.1 VideoInputTs Attributes

6.5.6.13.1.1 Index

This is the index for an instance of the VideoInputTs object.

6.5.6.13.1.2 Name

This is a unique name for this instance of the VideoInputTs object.

6.5.6.13.1.3 DecryptionEnabled

This attribute configures whether this input stream is encrypted for transport across the WAN. This WAN encryption is intended to be removed at the CCAP and not related to any CA encryption that may be configured for the output stream. A value of true means that the CCAP needs to decrypt this input stream as applicable. A value of false means that the CCAP does not need to decrypt this input stream. Default value is false.

6.5.6.14 UnicastVideoInputTs

This object specifies the unicast flow of an input transport stream.

Table 6.54: UnicastVideoInputTs Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
DestIpAddr	IpAddress	See attribute description			
DestUdpPort	InetPortNum	Yes			

A UnicastVideoInputTs object may be associated with a specific IpInterface. In this case, the DestIpAddr is not required. If an association is made to a UnicastVideoInputTsInterfaceName, care needs to be taken to make sure that the DestUdpPort specified does not overlap with the UDP port used for other traffic that may be present on the associated IpInterface instance.

Table 6.55: UnicastVideoInputTs Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
IpInterface	Association to IpInterface		0..1	UnicastVideoInputTsInterfaceName

6.5.6.14.1 UnicastVideoInputTs Object Attributes

6.5.6.14.1.1 DestIpAddr

This attribute corresponds to the IP destination address of the UDP IP flow of the input TS. This attribute is required unless the UnicastVideoInputTs object is associated with an IpInterface instance. If the IP address specified in the DestIpAddr attribute does not exist on the CCAP, the CCAP shall reject this configuration.

When the value of the DestIpAddr attribute is set to all zeros (e.g. 0.0.0.0), the CCAP shall listen for the traffic on the specified UDP port number on all IP interfaces.

6.5.6.14.1.2 DestUdpPort

This attribute corresponds to the UDP destination port of the UDP IP flow of the input TS.

6.5.6.15 MulticastVideoInputTs

This object specifies the multicast flows of an input transport stream. Having two MulticastVideoInputTs objects for one VideoInputTs occurs when input TS redundancy is configured (Hot-Hot sparing). If two MulticastVideoInputTs objects have the same Priority, this implies HOT-HOT redundancy. Which stream is actually forwarded is vendor-specific.

Table 6.56: MulticastVideoInputTs Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
SourceIpAddress	IpAddress	Yes (Key)			
GroupDestIpAddress	IpAddress	Yes (Key)			
DestUdpPort	InetPortNum	No			0
Priority	Byte	Yes			

A MulticastVideoInputTs object may be associated with a specific IpInterface. This associations provides a static mapping of the source of an input transport stream to an IP interface.

Table 6.57: MulticastVideoInputTs Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
IpInterface	Association to IpInterface		0..1	MulticastVideoInputTsInterfaceName

6.5.6.15.1 MulticastVideoInputTs Object Attributes

6.5.6.15.1.1 SourceIpAddress

This attribute corresponds to the Source Specific Multicast IP Address of the UDP IP flow.

6.5.6.15.1.2 GroupDestIpAdress

This attribute corresponds to the group address of a SSM (Source Specific Multicast) origination input TS.

6.5.6.15.1.3 DestUdpPort

This attribute corresponds to the UDP destination port of the UDP IP flow of the input TS.

6.5.6.15.1.4 Priority

This attribute is a number that identifies the preference order of this transport stream; higher number indicates a higher priority. It is used to order the multicast transport stream for the purpose of redundancy in the case of multiple multicast video sources. If two entries have the same "Priority", it implies Hot-Hot redundancy.

6.5.6.16 VideoOutputTs

The VideoOutputTs object represents a configuration multiplex of one or more ProgramSession, PidSession, or MptsPassThruSession instances.

Table 6.58: VideoOutputTs Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default
Index	UnsignedInt	Yes (Key)			
Name	String	No			""

Table 6.59: VideoOutputTs Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
VideoDownChannel	Directed aggregation to VideoDownChannel	1	1..*	

6.5.6.16.1 VideoOutputTs Object Attributes

6.5.6.16.1.1 Index

This is an index for an instance of this Object. It uniquely identifies a CCAP-generated MPTS composed of one or more program streams, PID streams and/or pass thru MPTS. This is NOT the Output TSID used for replication.

6.5.6.16.1.2 Name

This attribute configures the name of this instance of VideoOutputTs.

6.5.6.17 VideoDownChannel

This configuration object is included in figure 6.5 for reference. It is defined in clause 6.5.5.14.

6.5.6.18 DownChannel

This configuration object is included in figure 6.5 for reference. It is defined in clause 6.5.5.12.

6.5.6.19 ErmParams

This configuration object allows for the configuration of the needed parameters that are communicated to an ERM for a given DownChannel object instance.

Table 6.60: ErmParams Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
InputMapGroupName	String	Yes			
PhyLockParams	EnumBits	No	frequency(0), bandwidth(1), power(2), modulation(3), interleaver(4), j83Annex(5), symbolRate(6), mute(7)		"H
AllocationType	Enum	No	other(1), docsisOnly(2), videoOnly(3), any(4)		any

Table 6.61: ErmParams Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
EncryptionCapability	Directed composition to EncryptionCapability		0..3	
ErmRegistration	Directed aggregation to ErmRegistration		0..1	ErmRegistrationErmName

6.5.6.19.1 ErmParams Object Attributes

6.5.6.19.1.1 InputMapGroupName

This attribute represents the address field in the WithdrawnRoute and ReachableRoutes ERRP attributes. This attribute is optional for DocsisDownChannel.

6.5.6.19.1.2 PhyLockParams

This attribute represents the PHY parameters Lock state of the QAM channels for DEPI-initiated PHY parameters updates.

6.5.6.19.1.3 AllocationType

This attribute is an enumeration defining for which services this specific DownChannel instance can be allocated. A value of "any" means that the ERM could configure the QAM resource for either video or DOCSIS. The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.6.20 EncryptionCapability

The EncryptionCapability object defines one encryption option of the Encryptor that needs to be reported to the ERM. There can be up to three EncryptionCapability objects per QAM. In return, the ERM is expected to create dynamic sessions using one of the reported encryption options.

Table 6.62: EncryptionCapability Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes (Key)			
CaEncryptor	Enum	Yes	other(1), motorola(2), cisco(3), simulcrypt(4)		
EncryptionScheme	Enum	Yes	other(1), des(2), aes(3), 3des(4), dvbcsa(5), dvbcsa3(6)		
KeyLength	UnsignedInt	Yes			

6.5.6.20.1 EncryptionCapability Object Attributes

6.5.6.20.1.1 Index

This attributes assigns a unique identifier to this instance of the EncryptionCapability object.

6.5.6.20.1.2 CaEncryptor

This enumeration defines the type of CA encryption the Encryptor uses. The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.6.20.1.3 EncryptionScheme

This attribute defines the encryption algorithms applicable to the CA encryption defined by the CaEncryptor attribute. The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.6.20.1.4 KeyLength

This attribute defines the key length applicable to the algorithm defined by the EncryptionScheme attribute.

6.5.6.21 ErmRegistration

This object allows for the configuration of the interface to an Edge Resource Manager. Generally, one configured ERM interface exists for the entire CCAP. An ErmRegistration object contains the attributes in the following table. The CCAP may support only one instance of the ErmRegistration object. Configuring more than one ERM is generally used for scaling purposes, with each individual ERM being focused on specific, unique service groups. More than one ERM cannot be practically used to support the same service group, and there could be conflicts between the control messages of the independent ERMs.

Table 6.63: ErmRegistration Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
ErmName	String	Yes (Key)			
ErmAddress	Host	Yes			
ErmPort	InetPortNum	No			0
ErmConnectionType	Enum	No	other(1), client(2), server(3), clientAndServer(4)		client
HoldTimer	UnsignedInt	No	0 | 3.. 3 600	seconds	240
ConnRetryTimer	UnsignedInt	No		seconds	20
NextHopAddressDomain	UnsignedInt	Yes			
CompAddress	Host	Yes			
CompPort	InetPortNum	No			6 069
StreamingZone	String	Yes	1..255		
Id	UnsignedInt	No			0
Cost	UnsignedInt	No			0
CompName	String	Yes	1..255		

6.5.6.21.1 ErmRegistration Object Attributes

6.5.6.21.1.1 ErmName

This key represents the name of the ERM related to this registration instance. This is an internal reference for associating, e.g. QAM channels and input interfaces to an ERM.

6.5.6.21.1.2 ErmAddress

This attribute represents the IP Address or FQDN of the ERM.

6.5.6.21.1.3 ErmPort

This attribute represents the TCP port number used to reach the ERM.

6.5.6.21.1.4 ErmConnectionType

This attribute represents the type of TCP connection that is established by the CCAP. The value can be one of the following:

1) other(1) indicates that a vendor-extension has been implemented for this attribute.

2) client(2) indicates that the CCAP has to initiate the TCP connection with the ERM.

3) server(3) indicates that the CCAP has to wait for the TCP connection from the ERM.

4) clientAndServer(4) indicates that either the CCAP or the ERM can initiate the TCP connection.

6.5.6.21.1.5 HoldTimer

This attribute represents the number of seconds that the sender proposes for the value of the hold timer. Upon receipt of an OPEN message, the CCAP shall calculate the value of the Hold Timer by using the smaller of its configured Hold Time and the Hold Time received in the OPEN message.

The Hold Time has to be either zero or at least three seconds.

The CCAP may reject connections on the basis of the Hold Time. The calculated value indicates the maximum number of seconds that may elapse between the receipt of successive KEEPALIVE and/or UPDATE messages by the sender.

6.5.6.21.1.6 ConnRetryTimer

This attribute represents the time in seconds for the connect retry timer. The exact value of the connect retry timer is a local matter, but should be sufficiently large to allow TCP initialization.

6.5.6.21.1.7 NextHopAddressDomain

This attribute represents the address domain number of the next-hop server. The NextHopServer specifies the address to which a manager should use to connect to the component in order to control the resource specified in the reachable route. This parameter is used in the ERRP NextHopServer attribute.

6.5.6.21.1.8 CompAddress

This attribute represents the host portion of the ERRP NextHopServer attribute. This field contains an FQDN, or an IPv4 address using the textual representation defined in section 2.1 of RFC 1123 [i.7], or an IPv6 address using the textual representation defined in section 2.2 of RFC 4291 [i.16]. This value is sent in the ERRP NextHopServer attribute with the CompPort value in the ERRP messages. The attribute is optional when signaling DOCSIS only resources, however it is defined as a mandatory attribute since the typical use of ErmRegistration is for video.

6.5.6.21.1.9 CompPort

This attribute represents the port portion of the ERRP NextHopServer attribute. This field contains numerical value (1-65 535) representing the port number. If the port is empty or not given, the default port 6 069 is assumed. This value is sent in the ERRP NextHopServer attribute with the CompAddress value in the ERRP messages. The attribute is optional when signaling DOCSIS only resources, however it is defined as a mandatory attribute since the typical use of ErmRegistration is for video.

6.5.6.21.1.10 StreamingZone

This attribute represents the name of the Streaming Zone within which the component operates. This parameter is used in the ERRP OPEN message. StreamingZone Name is a mandatory parameter when supporting video applications. The capability is optional when signaling DOCSIS only resources.

The value is to be set to the string that represents the StreamingZone Name, i.e. <region>. The characters comprising the string are in the set within TEXT defined in section 15.1 of [i.8]. The CCAP shall support minimum string lengths of 64 for the StreamingZone attribute of the ErmRegistration object; however, the composition of the string used is defined by implementation agreements specified by the service provider.

A CCAP will exist in a single streaming zone.

6.5.6.21.1.11 Id RFC 4291 [i.16]

This attribute represents the unique identifier for the CCAP device within its Streaming Zone. This value can be set to the 4-octet value of an IPv4 address assigned to that device. This ID value is determined on startup and is the same for all peer connections. This parameter is used in the ERRP OPEN message header.

6.5.6.21.1.12 Cost

This attribute represents the static cost for use of this resource.

6.5.6.21.1.13 CompName

The name of the component for which the data in the update message applies. This parameter is used in the ERRP OPEN message. Component Name is a mandatory parameter when supporting video applications. The capability is optional when signaling DOCSIS only resources.

The value is to be set to the string that represents the Component Name, i.e. <region>.<localname>. The characters comprising the string are in the set within TEXT defined in section 15.1 of RFC 2326 [i.8]. The CCAP shall support minimum string lengths of 64 for the CompName attribute of the ErmRegistration object; however, the composition of the string used is defined by implementation agreements specified by the service provider.

6.5.6.22 VideoSession

The VideoSession abstract object holds the common attributes for the session configuration objects (program, PID, and MPTS passthrough).

Table 6.64: VideoSession Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes (Key)			
Name	String	No	0..32		""

Table 6.65: VideoSession Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
VideoInputTs	Directed aggregation to VideoInputTs	0..*		VideoInputTs
VideoOutputTs	Association to VideoOutputTS	0..*	1..*	VideoOutputTsIndex

6.5.6.22.1 VideoSession Object Attributes

6.5.6.22.1.1 Index

This is the index for the configured session.

6.5.6.22.1.2 Name

This attribute is the name of the session. Unique names are given to each instance of a session type.

6.5.6.23 ProgramSession

The ProgramSession object allows the identification, encryption, processing and insertion of a single program stream into a VideoOutputTs. Each ProgramSession object needs to have one VideoInputTs object associated with it.

Table 6.66: ProgramSession Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
InputMpegProgramNum	UnsignedShort	Yes			
OutputMpegProgramNum	UnsignedShort	Yes			
PatPidRemap	Boolean	No			true
RequestedBandwidth	UnsignedInt	Yes		bps	

To define a ProgramSession object, it needs to specify either a "unicast" or a "multicast" TSVideoInput object.

Table 6.67: ProgramSession Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
VideoSession	Specialization of VideoSession			
CasInfo	Directed aggregation to CasInfo	0..*	0..1	CasInfoIndex
EncryptionData	Directed aggregation to EncryptionData	0..*	0..1	EncryptionDataIndex
EncryptControl	Directed aggregation to EncryptControl	0..*	0..1	EncryptControlIndex

6.5.6.23.1 ProgramSession Object Attributes

6.5.6.23.1.1 InputMpegProgramNum

This attribute selects a specific program from the transport stream of the incoming video stream. This program number should be part of the incoming PAT. A value of 0 (zero) means that any incoming program number can be accepted.

6.5.6.23.1.2 OutputMpegProgramNum

This attribute specifies the program number to be present in the transport stream of the outgoing video stream. This program number will be part of the outgoing PAT.

6.5.6.23.1.3 PatPidRemap

A value of true indicates that the elementary stream PID of this input program can be remapped to the VideoOutputTs, as long as the PAT and PMT are updated. A value of false indicates that the same input elementary stream PID has to be used on the VideoOutputTs.

6.5.6.23.1.4 RequestedBandwidth

This attribute configures the expected bandwidth parameters for a static input video session described by this object. This parameter is used for encryption and video down channel output resources. A value of 0 (zero) means that no bandwidth validation is required.

6.5.6.24 MptsPassThruSession

The MptsPassThruSession object allows the identification and insertion of an unmodified MPTS into a VideoOutputTs. Each MptsPassThruSession object needs to have one VideoInputTs object associated with it; this association is inherited through the abstract object VideoSession.

To define an MptsPassThruSession object, specify either a "unicast" or a "multicast" VideoInputTs object.

Table 6.68: MptsPassThruSession Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
VideoSession	Specialization of VideoSession			

6.5.6.25 PidSession

The PidSession object allows the identification, processing and insertion of a PID stream into a VideoOutputTs. Each PidSession object needs to have one VideoInputTs object associated with it.

Table 6.69: PidSession Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
InputPid	HePidValue	Yes	0..8 191 | 65 535		
PidRemapEnable	Boolean	No			false
PidType	Enum	Yes	other(1), emm(2), nit(3), cat(4), pat(5), fixed(6), eas(7), dsm-cc(8), eiss(9), etvbif(10), video(11), audio(12)		
CasId	HexBinary	No	size(8)		00000000

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
OutputPid	HePidValue	Yes			
OutputProgramNumber	UnsignedShort	No			

Table 6.70: PidSession Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
VideoSession	Specialization of VideoSession			

6.5.6.25.1 PidSession Object Attributes

6.5.6.25.1.1 InputPid

This attribute identifies a specific PID stream in the input transport stream.

6.5.6.25.1.2 PidRemapEnable

This object configures whether or not the identified PID stream can be remapped when inserted in the VideoOutputTs.

6.5.6.25.1.3 PidType

This enumeration defines the type of the identified PID stream. This value is used to understand what anchor table (i.e. PAT, CAT) would need to be updated in case PidRemapEnable is set to True and a remap is required. In case of type "eas", the table sections of the PID stream may need to be interleaved with other table sections that would be present on the same OutputPid. "dsm-cc" is used for digital storage media command and control. "eiss" is used for ETV Integrated Signaling Streams (Stream type 0xC0 or 0x05 w-descriptor tag 0xA2). "etvbif" is used for ETV Binary Interchange Format (Stream type 0xC0 or 0x05 w-descriptor tag 0xA1 OR Stream Type 0X0B). "video" is used for MPEG2 video streams. "audio" is used for MPEG2 audio streams. The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.6.25.1.4 CasId

This attribute allows a proper identification of the CAT table parameter(s) that need(s) to be updated when the PidType is set to "EMM", PidRemapEnable is set to True and a remap is required. This parameter is required in Simulcrypt operation when the CAT advertises more than one EMM PID streams. A value of 0 means that no CAS ID is associated with this PID Session.

6.5.6.25.1.5 OutputPid

This attribute defines the expected PID value of the identified PID stream when inserted in the VideoOutputTS. However, the OutputPid value cannot be guaranteed if the PidRemapEnable flag is set to True.

6.5.6.25.1.6 OutputProgramNumber

This attribute defines the output program number for the PID session.

6.5.6.26 Chassis

This configuration object is included in figure 6.5 for reference. It is defined in clause 6.5.5.2.

6.5.6.27 Decryptor

The Decryptor object provides for the configuration of a Decryptor module or modules in the CCAP that are used to decrypt encrypted content delivered to the CCAP.

Table 6.71: Decryptor Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes (Key)			
CwTimeout	UnsignedInt	No		seconds	10

The Decryptor object has the following association.

Table 6.72: Decryptor Object Associations

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

6.5.6.27.1 Decryptor Object Attributes

6.5.6.27.1.1 Index

The Index is an unsigned, 32-bit identifier used as a key for this object.

6.5.6.27.1.2 CwTimeout

This attribute configures the length of time in seconds that the Decryptor should wait for an ECM Decoder (ECMD) before switching to a redundant unit.

6.5.6.28 EcmdUsage

The EcmdUsage object provides for the configuration of multiple decryption sessions. It is an intermediate object that provides linkages between Decryptor objects and the ECMD(s) associated with those encrypted streams. The ECMD object is defined in clause 6.5.6.29.

Table 6.73: EcmdUsage Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes (Key)			
Priority	UnsignedInt	Yes			

The EcmdUsage object has the following association.

Table 6.74: EcmdUsage Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
Ecmd	Directed aggregation to Ecmd	1..*	1	EcmdIndex

6.5.6.28.1 EcmdUsage Object Attributes

6.5.6.28.1.1 Index

This is an index for an instance of this Object. The EcmdUsage object is a pointer to the Ecmd object that can be used for any program session that requires decryption as long as the CAS identifier of the input program matches.

6.5.6.28.1.2 Priority

This is the configured selection priority for any program session that requires decryption when multiple ECMDs with the same CAS identifier are active. The ECMD with the lowest number should be selected first.

6.5.6.29 Ecmd

This object allows for the configuration of the interface to an Entitlement Control Message Decoder (ECMD).

Table 6.75: Ecmd Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
NumberDecryptedStreams	UnsignedInt	Yes			

Table 6.76: Ecmd Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
Ecm	Specialization of Ecm			

6.5.6.29.1 Ecmd Object Attributes

6.5.6.29.1.1 NumberDecryptedStreams

The maximum number of decrypted streams the ECMD should handle.

6.5.6.30 Ecm

This abstract object holds the common attributes of ECMD and ECMG instances.

Table 6.77: Ecm Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes (Key)			
Server	Host	Yes			
ServerPort	InetPortNum	Yes			
CasId	HexBinary	Yes	size(8)		

6.5.6.30.1 Ecm Object Attributes

6.5.6.30.1.1 Index

The Index is an unsigned, 32-bit identifier used as a key for this object.

6.5.6.30.1.2 Server

This is the IP address or FQDN of the ECMD/ECMG. Encryption code words are sent to this address and ECMs are received from this address.

6.5.6.30.1.3 ServerPort

This is the far-end TCP port for communication.

6.5.6.30.1.4 CasId

This attribute defines the CA System ID that the ECMD/ECMG will use.

6.5.6.31 Slot

This configuration object is included in figure 6.5 for reference. It is defined in clause 6.5.5.4.

6.5.6.32 LineCard

This configuration object is included in figure 6.5 for reference. It is defined in clause 6.5.5.5.

6.5.6.33 RfLineCard

This configuration object is included in figure 6.5 for reference. It is defined in clause 6.5.5.6.

6.5.6.34 Encryptor

This object allows for the configuration of an Encryptor. Each Encryptor object is part of a DLC. Each can be associated with one active and zero or more backup ECMGs. For Simulcrypt, the Encryptor would be associated with multiple active ECMGs, each for a different CAS. An Encryptor object contains the attributes in table 6.78.

Table 6.78: Encryptor Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes (Key)			
CaEncryptorType	Enum	Yes	other(1), motorola(2), cisco(3), simulcrypt(4),	NA	
ClearStreamTimeout	UnsignedShort	No		seconds	10
EcmTimeout	UnsignedShort	No		seconds	10

Encryptor has the following association.

Table 6.79: Encryptor Object Associations

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

6.5.6.34.1 Encryptor Object Attributes

6.5.6.34.1.1 Index

This is an index for an instance of this object.

6.5.6.34.1.2 CaEncryptorType

This enumeration defines the type of CA encryption the Encryptor uses. The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.6.34.1.3 ClearStreamTimeout

This configured attribute defines the number of seconds a given stream may be sent in the clear when the stream is configured to be encrypted. If this timer expires and the session has not received any encryption information from the ECMG, the CCAP shall discontinue forwarding this stream.

6.5.6.34.1.4 EcmTimeout

This attribute configures the number of seconds that a CCAP will wait to get a response from a ECMG before switching to the redundant unit.

6.5.6.35 EcmgUsage

The EcmgUsage object provides for the configuration of multiple encryption sessions. It is an intermediate object that provides linkages between Encryptor objects and the ECMG(s) associated with those encrypted streams. The ECMG object is defined in clause 6.5.6.36.

Table 6.80: EcmgUsage Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes (Key)			
Priority	UnsignedInt	Yes			

The EcmgUsage object has the following association.

Table 6.81: EcmgUsage Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
Ecmg	Directed aggregation to Ecmg	1..*	1	EcmgIndex

6.5.6.35.1 EcmgUsage Object Attributes

6.5.6.35.1.1 Index

This is an index for an instance of this object. It is a pointer to an active Ecmg object that can be used for any program session that requires encryption as long as the CAS identifier matches.

6.5.6.35.1.2 Priority

This is the configured selection priority for any program session that requires encryption when multiple ECMGs with the same CAS identifier are active. The ECMG with the lowest number should be selected first.

6.5.6.36 Ecmg

This object allows for the configuration of the interface to an Entitlement Control Message Generator (ECMG). Redundant ECMGs for the same CAS may exist, each with the same CA_System_ID, with the priority determining which is currently in use by an Encryptor for a particular CAS. An Ecmg object contains the attributes in table 6.82.

Table 6.82: Ecmg Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
RecommendedCpDuration	UnsignedInt	Yes	1..*	seconds	
NumberEncryptedStreams	UnsignedInt	Yes		streams	

Table 6.83: Ecmg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
Ecm	Specialization of Ecm			

6.5.6.36.1 Ecmg Object Attributes

6.5.6.36.1.1 RecommendedCpDuration

The recommended cryptoperiod, in seconds, the ECMG should expect.

6.5.6.36.1.2 NumberEncryptedStreams

The maximum number of encrypted streams the ECMG should handle.

6.5.6.37 StaticUdpMapEncryption

This object allows for the configuration of encryption for all static UDP port-mapped sessions on a given downstream RF line card. When this object is associated with an RfLineCard instance, all static UDP port-mapped sessions on that RF Line Card are configured for encryption per the associated encryption objects (the mandatory objects of EncryptControl and CasInfo, and the optional object EncryptionData).

If the StaticUdpMapEncryption object is configured without an association to an instance of EncryptControl or CasInfo, the CCAP shall reject the configuration instance.

Since this functionality is not used by all operators, implementation of this configuration object in the CCAP is not mandatory; the CCAP may exclude this configuration object.

Table 6.84: StaticUdpMapEncryption Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	Integer	Yes			

Table 6.85: StaticUdpMapEncryption Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
EncryptControl	Directed aggregation to EncryptControl	0..*	0..1	EncryptControlIndex
CasInfo	Directed aggregation to CasInfo	0..*	0..1	CasInfoIndex
EncryptionData	Directed aggregation to EncryptionData	0..*	0..1	EncryptionDataIndex

6.5.6.37.1 StaticUdpMapEncryption Object Attributes

6.5.6.37.1.1 Index

This attribute configures a unique index for an instance of this object.

6.5.7 DOCSIS® Configuration Objects

The objects in the following sections configure DOCSIS on the CCAP. They have been grouped logically for usability.

6.5.7.1 DOCSIS® System Configuration

These objects define global DOCSIS configuration for the CCAP.

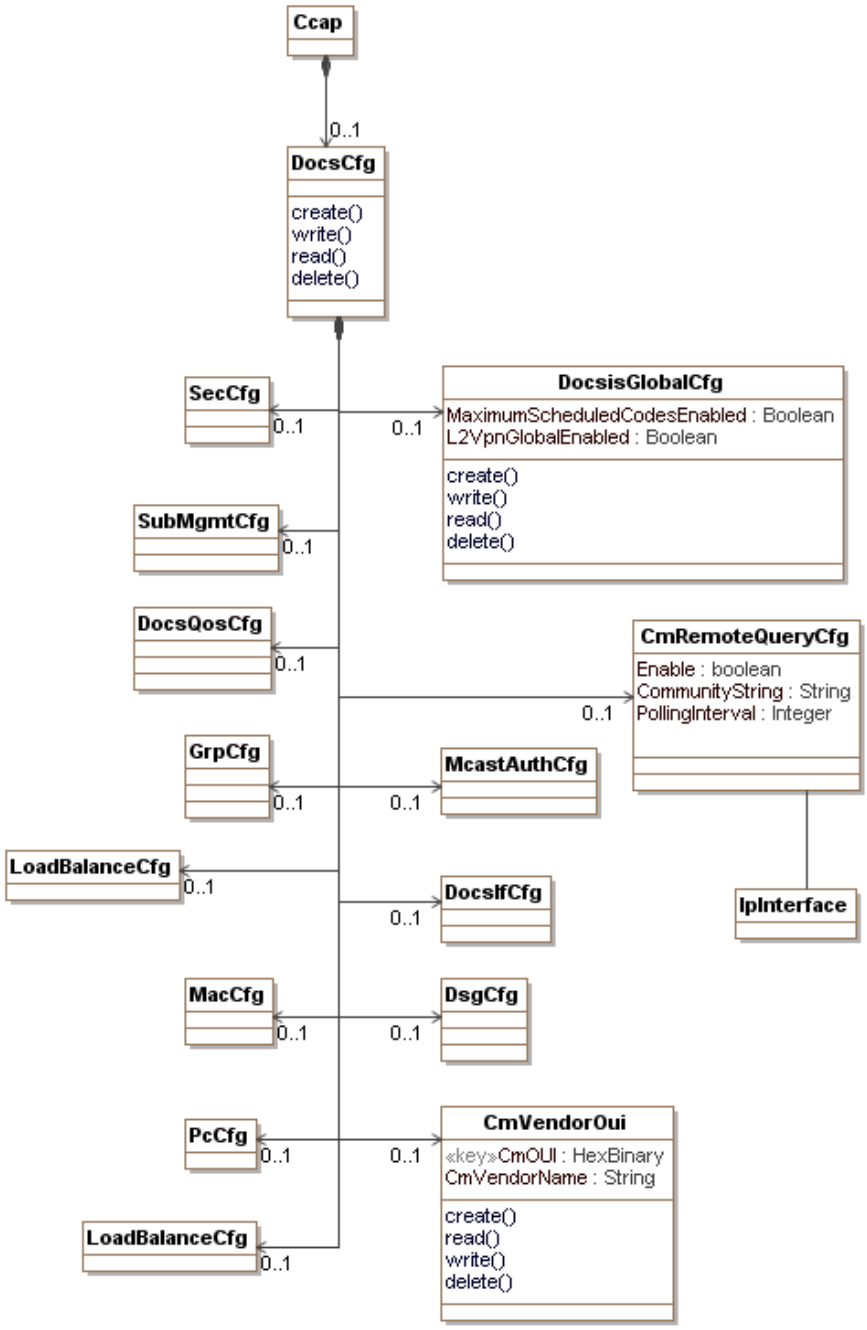


Figure 6.6: DOCSIS® Configuration Objects

6.5.7.1.1 Ccap

This configuration object is included in figure 6.6 for reference. It is defined in clause 6.5.4.1.

6.5.7.1.2 DocsCfg

The DocsCfg object is the primary container of DOCSIS configuration objects. It has the following associations:

Table 6.86: DocsCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
SecCfg	Directed composition to SecCfg		0..1	
SubMgmtCfg	Directed composition to SubMgmtCfg		0..1	
DocsQosCfg	Directed composition to DocsQosCfg		0..1	
GrpCfg	Directed composition to GrpCfg		0..1	
MacCfg	Directed composition to MacCfg		0..1	
PcCfg	Directed composition to PcCfg		0..1	
LoadBalanceCfg	Directed composition to LoadBalanceCfg		0..1	
DocsisGlobalCfg	Directed composition to DocsisGlobalCfg		0..1	
CmRemoteQuery	Directed composition to CmRemoteQuery		0..1	
McastAuthCfg	Directed composition to McastAuthCfg		0..1	
DocsIfCfg	Directed composition to DocsIfCfg		0..1	
DsgCfg	Directed composition to DsgCfg		0..1	
CmVendorOUI	Directed composition to CmVendorOUI		0..1	

6.5.7.1.3 SecCfg

This configuration object is included in figure 6.6 for reference. It is defined in clause 6.5.7.2.3.

6.5.7.1.4 SubMgmtCfg

This configuration object is included in figure 6.6 for reference. It is defined in clause 6.5.7.3.3.

6.5.7.1.5 DocsQosCfg

This configuration object is included in figure 6.6 for reference. It is defined in clause 6.5.7.4.2.

6.5.7.1.6 GrpCfg

This configuration object is included in figure 6.6 for reference. It is defined in clause 6.5.7.5.3.

6.5.7.1.7 MacCfg

This configuration object is included in figure 6.6 for reference. It is defined in clause 6.5.7.6.3.

6.5.7.1.8 PcCfg

This configuration object is included in figure 6.6 for reference. It is defined in clause 6.5.7.10.2.

6.5.7.1.9 LoadBalanceCfg

This configuration object is included in figure 6.6 for reference. It is defined in clause 6.5.7.11.

6.5.7.1.10 DocsisGlobalCfg

The DocsisGlobalCfg object defines DOCSIS configuration attributes for the entire system, such as enabling Maximum Scheduled Codes and L2VPN.

Table 6.87: DocsisGlobalCfg Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
MaximumScheduledCodesEnabled	Boolean	Yes			
L2VpnGlobalEnabled	Boolean	No			false

6.5.7.1.10.1 DocsisGlobalCfg Object Attributes

6.5.7.1.10.1.1 MaximumScheduledCodesEnabled

Indicates the global state of the Maximum Scheduled Code feature on the CCAP. The value true indicates that this feature can be enabled on individual logical channels on the CCAP. The value false indicates that the feature is not in operation on the CCAP. Note that the CCAP object attribute ScdmaChannelMscState enables or disables Maximum Scheduled Codes on a per logical channel basis.

6.5.7.1.10.1.2 L2VpnGlobalEnabled

This attribute will enable or disable on a global basis the configuration of L2VPN forwarding for all DOCSIS MAC domains. The default value is false. This attribute only enables L2VPN forwarding; configuration of the feature is handled in a vendor-specific way.

Reference: [7], Annex H, docsIfExt2CmtsMscGlobalEnable section.

6.5.7.1.11 McastAuthCfg

This configuration object is included in figure 6.6 for reference. It is defined in clause 6.5.7.7.3.

6.5.7.1.12 DocsIfCfg

This configuration object is included in figure 6.6 for reference. It is defined in clause 6.5.7.8.2.

6.5.7.1.13 DsgCfg

This configuration object is included in figure 6.6 for reference. It is defined in clause 6.5.7.9.2.

6.5.7.1.14 CMRemoteQuery

This configuration object enables SNMP queries of CMs.

Table 6.88: CmRemoteQuery Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Enable	Boolean	Yes			
CommunityString	String	Yes			
PollingInterval	Integer	Yes		Seconds	

This object is associated with the source interface address on the CCAP.

Table 6.89: CmRemoteQuery Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
IpInterface	Association to IpInterface			

6.5.7.1.14.1 CmRemoteQuery Object Attributes

6.5.7.1.14.1.1 Enabled

This attribute configures whether or not CM remote query is enabled on the CCAP.

6.5.7.1.14.1.2 CommunityString

This attribute configures the SNMP Community String for remote queries.

6.5.7.1.14.1.3 PollingInterval

This attribute configures the minimum amount of time in seconds between consecutive polls of the same MIB object on the same cable modem.

6.5.7.1.15 CmVendorOui

This configuration object allows the operator to create a database of OUIs and Vendors.

Table 6.90: CmVendorOui Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
CmOUI	HexBinary	Yes (Key)	size(3)		
CmVendorName	String	Yes			

6.5.7.1.15.1 CmVendorOui Object Attributes

6.5.7.1.15.1.1 CmOUI

This attribute configures the OUI portion of a given MAC address.

6.5.7.1.15.1.2 CmVendorName

This attribute configures the company name of the vendor with the associated OUI.

6.5.7.2 DOCSIS® Security Configuration

This section details the DOCSIS configuration objects for Security features defined in DOCSIS 3.0. These objects have been modified from [7] to remove the SMIv2 and SNMP attributes from the configured objects and attributes. The object model for these features is shown in figure 6.7.

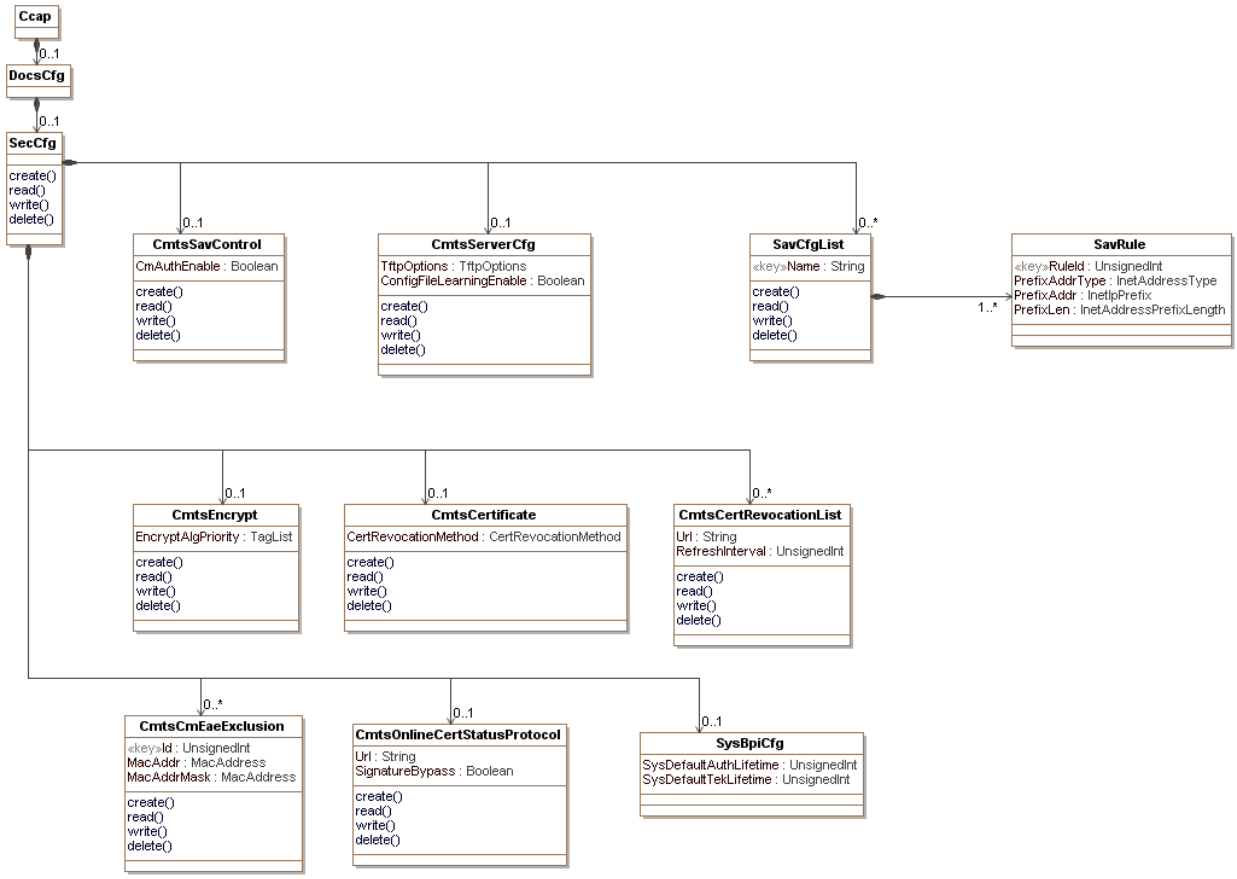


Figure 6.7: DOCSIS® Security Configuration Objects

6.5.7.2.1 Ccap

This configuration object is included in figure 6.7 for reference. It is defined in clause 6.5.4.1.

6.5.7.2.2 DocsCfg

This configuration object is included in figure 6.7 for reference. It is defined in clause 6.5.7.1.2.

6.5.7.2.3 SecCfg

The SecCfg object is the primary container of DOCSIS security configuration objects. It has the following associations:

Table 6.91: SecCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
SavCfgList	Directed composition to SavCfgList		0..*	
CmtsSavControl	Directed composition to CmtsSavControl		0..1	
CmtsServerCfg	Directed composition to CmtsServerCfg		0..1	
CmtsEncrypt	Directed composition to CmtsEncrypt		0..1	
CmtsCertificate	Directed composition to CmtsCertificate		0..1	
CmtsCertRevocationList	Directed composition to CmtsCertRevocationList		0..*	
CmtsCmEaeExclusion	Directed composition to CmtsCmEaeExclusion		0..*	
CmtsOnlineCertStatusProtocol	Directed composition to CmtsOnlineCertStatusProtocol		0..1	
SysBpiCfg	Directed composition to SysBpiCfg		0..1	

6.5.7.2.4 SavCfgList

This configuration object allows for the configuration of a Source Address Verification(SAV) list which can contain one or more rules for the Prefixes that are managed by this group.

This object supports the creation and deletion of multiple instances. Each object instance defines one CMTS SAV list that will contain 1 or more SAV rules. The SavRule Object will provide the configuration of each of the configured subnet prefix extension for which the CCAP performs source address verification.

Creation of a new instance of this object requires the Name attribute to be set.

Reference: [7], DOCS-SEC-MIB section.

Table 6.92: SavCfgList Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Name	String	Yes (Key)			

Table 6.93: SavCfgList Object Associations

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

6.5.7.2.4.1 SavCfgList Object Attributes

6.5.7.2.4.1.1 Name

This attribute is the key that identifies the instance of the SavCmAuth object to which this object extension belongs.

6.5.7.2.5 SavRule

This configuration object consists of the read-write objects of the docsSecSavCfgListEntry defined in [7] and will be used without further modification for CCAP.

This object supports the creation and deletion of multiple instances. Each object instance defines one CMTS configured subnet prefix extension for which the CCAP performs source address verification.

Creation of a new instance of this object requires the PrefixAddrType and PrefixAddr attributes to be set.

Reference: [7], DOCS-SEC-MIB section.

6.5.7.2.6 CmtsSavControl

This configuration object is based on the CmtsSavCtrl object defined in Annex L of [7] and will be used without modification for CCAP.

This object defines attributes for global Source Address Verification (SAV) configuration.

Reference: [7], CmtsSavCtrl Object section.

6.5.7.2.7 CmtsServerCfg

This configuration object is based on the CmtsServerCfg object defined in Annex L of [7] and will be used without modification for CCAP.

This object defines attributes for configuring TFTP Configuration File Security features.

Reference: [7], CmtsServerCfg Object section.

6.5.7.2.8 CmtsEncrypt

This configuration object is based on the CmtsEncrypt object defined in Annex L of [7] and will be used without modification for CCAP.

This object includes an attribute that defines the order in which encryption algorithms are to be applied.

Reference: [7], CmtsEncrypt Object section.

6.5.7.2.9 CmtsCertificate

This configuration object is based on the CmtsCertificate object defined in Annex L of [7] and will be used with the following modification for CCAP: An enumeration of other(1) has been added to the CertRevocationMethod enumeration to allow for vendor extension. The enumeration definitions can be found in table 6.94.

This object defines attributes for global certificate revocation configuration.

Reference: [7], CmtsCertificate Object section.

Table 6.94: CmtsCertificate Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
CertRevocationMethod	Enum	No	other(1), none(2), crl(3), ocsp(4), crlAndOcsp(5)		none

6.5.7.2.10 CmtsCertRevocationList

This configuration object consists of the read-write objects under the CmtsCertRevocationList object identifier defined in Annex L of [7]. The LastUpdate attribute has been removed.

This object defines a CRL location URL and periodic refresh interval value. The CRL location URL defines from where the CCAP will retrieve the CRL file. The periodic refresh interval value indicates how often the CCAP will retrieve the CRL file for updates if the tbsCertList.nextUpdate attribute in the file is absent.

This object is only applicable when the CertRevocationMethod attribute of the DocsSecCmtsCertificate object is set to "crl" or "crlAndOcsp".

Reference: [7], CmtsCertRevocationList Object section.

6.5.7.2.11 CmtsCmEaeExclusion

This configuration object consists of the read-write objects of the CmtsCmEaeExclusion object defined in Annex L of [7] and will be used with no further modifications for CCAP.

This object defines a list of CMs or CM groups to exclude from Early Authentication and Encryption (EAE). This object allows overrides to the value of EAE Control for individual CMs or group of CMs for purposes such as debugging.

The CCAP shall support a minimum of 30 instances of the CmtsCmEaeExclusion object.

This object is only applicable when the EarlyAuthEncryptCtrl attribute of the MdCfg object is enabled.

This object supports the creation and deletion of multiple instances.

Reference: [7], CmtsCmEaeExclusion Object section.

6.5.7.2.12 CmtsOnlineCertStatusProtocol

This configuration object is based on the CmtsOnlineCertStatusProtocol object defined in Annex L of [7] and will be used without modification for CCAP.

This object contains an OCSP Responder URL and an attribute to bypass signature checking of the OCSP response, as detailed in [10]. The CCAP will use the URL for OCSP communications in checking a certificate's revocation status. This object is only applicable when the CertRevocationMethod attribute of the CmtsCertificate object is set to "ocsp" or "crlAndOcsp".

Reference: [7], CmtsOnlineCertStatusProtocol Object section.

6.5.7.2.13 SysBpiCfg

This object is based on the DocsBpiCmtsBaseEntry table defined in RFC 3083 [12].

This object provides the configuration of the default Baseline Privacy key lifetimes. If not configured, the default values are vendor specific.

Reference: RFC 3083 [12].

6.5.7.3 DOCSIS® Subscriber Management Configuration

This group of configuration elements allows for the configuration of the Subscriber Management rules. They are based on the configuration elements from [7]. The configuration specific object model is shown in figure 6.8.

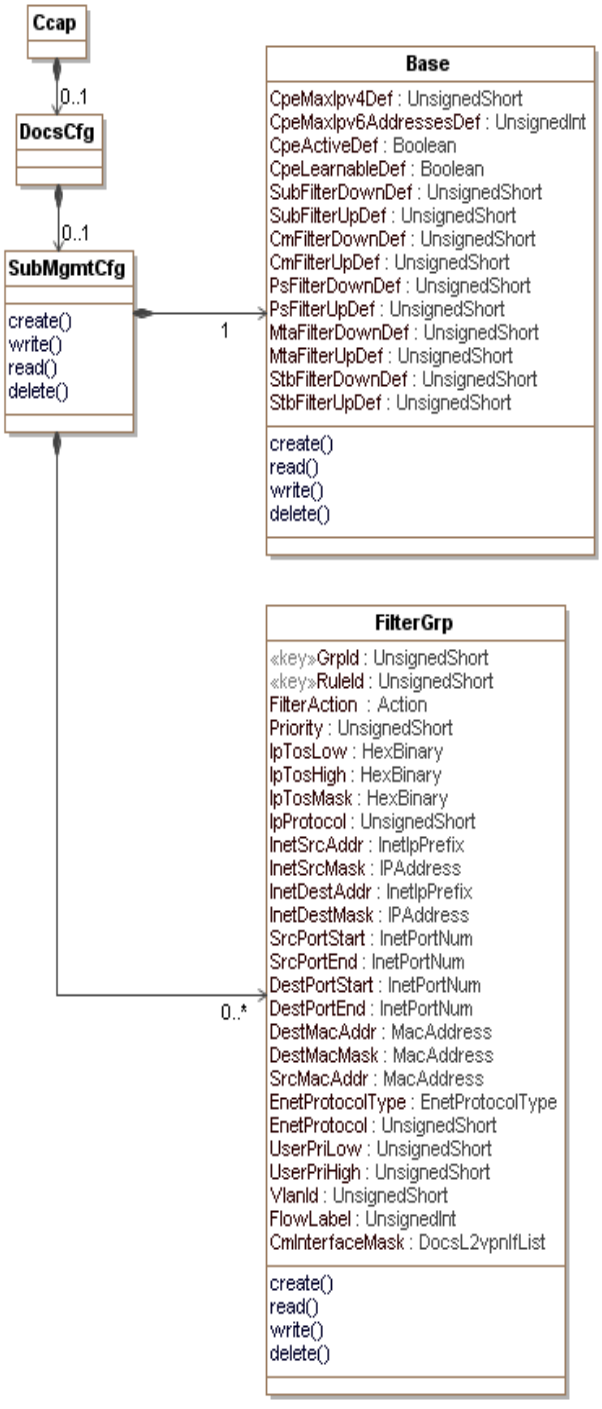


Figure 6.8: DOCSIS® Subscriber Management Configuration Objects

6.5.7.3.1 Ccap

This configuration object is included in figure 6.8 for reference. It is defined in clause 6.5.4.1.

6.5.7.3.2 DocsCfg

This configuration object is included in figure 6.8 for reference. It is defined in clause 6.5.7.1.2.

6.5.7.3.3 SubMgmtCfg

The SubMgmtCfg object is the primary container of DOCSIS security configuration objects. It has the following associations:

Table 6.95: SubMgmtCfg Object Associations

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

6.5.7.3.4 Base

This configuration object is based on the Subscriber Management Base object defined in [7] and will be used without modification for CCAP.

This object defines the configuration parameters of Subscriber Management features for the CM in case the CM does not signal any of the parameters during the registration process.

Reference: [7], Base Object section.

6.5.7.3.5 FilterGrp

This configuration object is based on the FilterGrp object defined in [7] and will be used with the following modifications for CCAP:

- The ClassPkts read-only attribute has been removed.
- The InetAddrType attribute has been removed.
- The Action attribute has been renamed FilterAction.
- An enumeration of other(1) has been added to the FilterAction and EnetProtocolType enumerations to allow for vendor extension. The enumeration definitions can be found in table 6.96.

This object describes a set of filter or classifier criteria. Classifiers are assigned by group to the individual CMs. That assignment is made via the "Subscriber Management TLVs" encodings sent upstream from the CM to the CCAP during registration, or in their absence, default values configured in the CCAP.

A Filter Group ID (GrpId) is a set of rules that correspond to the expansion of a UDC Group ID into individual UDC rules. The UDC Group IDs are linked to IDs of the FilterGrp object so the CCAP can signal those filter rules as UDCs to the CM during the registration process. Implementation of L2 classification criteria is optional for the CCAP; LLC/MAC upstream and downstream filter criteria can be ignored during the packet matching process.

Reference: [7], FilterGrp Object section.

Table 6.96: FilterGrp Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
FilterAction	Enum	No	other(1), permit(2), deny(3)		permit
EnetProtocolType	Enum	No	other(1), none(2), ethertype(3), dsap(4), mac(5), all(6)		ethertype

6.5.7.4 DOCSIS® QoS Configuration

This group of configuration elements allows for the configuration of DOCSIS QoS. The configuration specific object model is shown in figure 6.9.

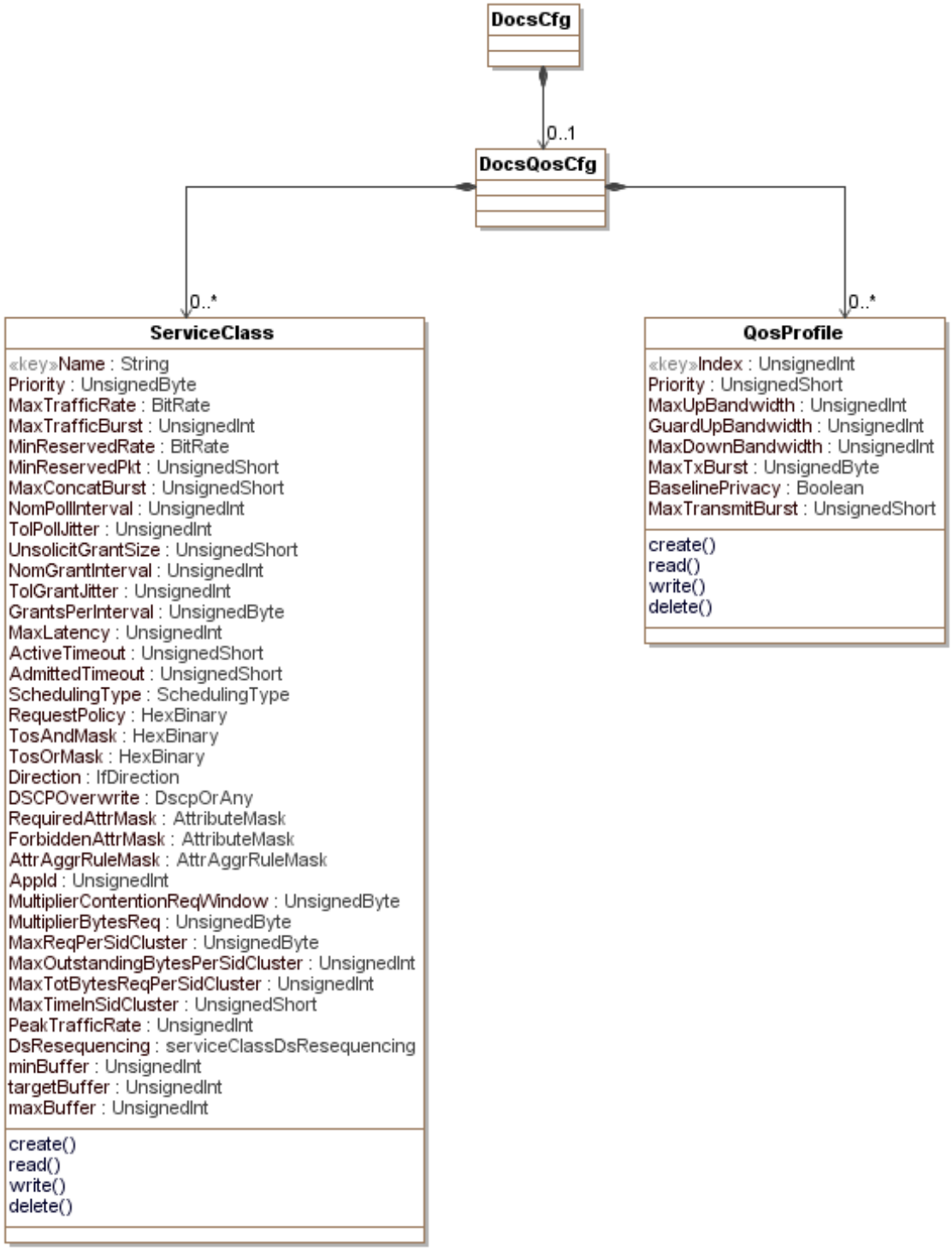


Figure 6.9: DOCSIS® QoS Configuration Objects

6.5.7.4.1 DocsCfg

This configuration object is included in figure 6.9 for reference. It is defined in clause 6.5.7.1.2.

6.5.7.4.2 DocsQosCfg

The DocsQosCfg object is the primary container of DOCSIS QOS configuration objects. It has the following associations:

Table 6.97: DocsQosCfg Object Associations

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

6.5.7.4.3 ServiceClass

This configuration object is based on the ServiceClass object defined in Annex O of [7]; the following modifications have been made:

- The StorageType attribute has been removed.
- An enumeration of other(1) has been added to the SchedulingType, Direction and DsResequencing enumerations to allow for vendor extension. The enumeration definitions can be found in the following attributes table.

This object describes a provisioned service class on a CCAP. Each object instance defines a template for certain DOCSIS QOS Parameter Set values. When a CM creates or modifies an Admitted QOS Parameter Set for a Service Flow, it may reference a Service Class Name instead of providing explicit QOS Parameter Set values. In this case, the CCAP populates the QOS Parameter Set with the applicable corresponding values from the named Service Class. Subsequent changes to a Service Class row do not affect the QOS Parameter Set values of any service flows already admitted. A service class template applies to only a single direction, as indicated in the ServiceClassDirection attribute.

Reference: [7], ServiceClass section.

Table 6.98: ServiceClass Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
SchedulingType	Enum	No	other(1), bestEffort(2), nonRealTimePollingService(3), realTimePollingService(4), unsolicitedGrantServiceWithAD(5), unsolicitedGrantService(6)		bestEffort
Direction	Enum	No	other(1), downstream(2), upstream(3)		upstream
DsResequencing	Enum	No	other(1), resequencingDcid(2), noResequincingDcid(3)		resequencingDcid

6.5.7.4.4 QosProfile

This configuration object consists of the read-write objects of the docsIfQosProfileTable defined in RFC 4546 [20] and is used with modifications for CCAP. The following attributes have been removed:

- Status
- StorageType

The QosProfile object is used to help provide a mapping between cable modems that have registered with a DOCSIS 1.0 style Class of Service. The support for this configuration is dependent on the CCAP supporting DOCSIS 1.0 style configuration files and CM registrations.

Reference: RFC 4546 [20], docsIfQosProfileTable.

6.5.7.5 DOCSIS® Multicast QoS Configuration

This group of configuration elements allows for the configuration of DOCSIS Multicast QoS. They are based on the configuration elements from [7]. The configuration specific object model is shown in figure 6.10.

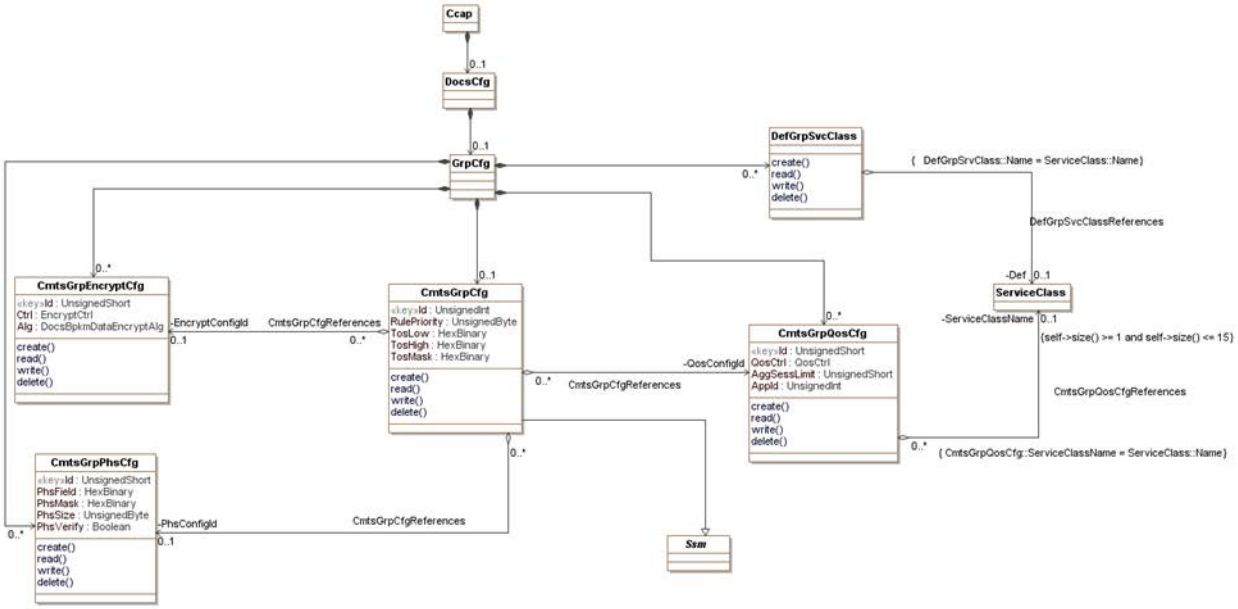


Figure 6.10: DOCSIS® Multicast QoS Configuration Objects

6.5.7.5.1 Ccap

This configuration object is included in figure 6.10 for reference. It is defined in clause 6.5.4.1.

6.5.7.5.2 DocsCfg

This configuration object is included in figure 6.10 for reference. It is defined in clause 6.5.7.1.2.

6.5.7.5.3 GrpCfg

The GrpCfg object is the primary container of DOCSIS Multicast QoS configuration objects. It has the following associations:

Table 6.99: GrpCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
CmtsGrpCfg	Directed composition to CmtsGrpCfg		0..1	
DefGrpSvcClass	Directed composition to DefGrpSvcClass		0..*	
CmtsGrpQosCfg	Directed composition to CmtsGrpQosCfg		0..*	
CmtsGrpEncryptCfg	Directed composition to CmtsGrpEncryptCfg		0..*	
CmtsGrpPhsCfg	Directed composition to CmtsGrpPhsCfg		0..*	

6.5.7.5.4 CmtsGrpCfg

This configuration object consists of the read-write objects of the CmtsGrpCfg object defined in Annex M of [7], and will be used with the following modifications for CCAP:

- The PrefixAddrType attribute has been removed.
- The following attributes have been moved into the abstract class Ssm:
  - SrcPrefixAddr
  - SrcPrefixLen
  - GrpPrefixAddr
  - GrpPrefixLen

This object controls the QoS, PHS, and encryption settings for downstream forwarding of IP multicast sessions. An IP multicast session is replicated to one or more Downstream Channel Sets (DCSs), where each DCS is either a single downstream channel or a downstream bonding group of multiple channels. The CCAP determines on which DCSs to replicate a multicast session based on IP multicast membership reports ("joins") or other vendor-specific static configuration.

The CmtsGrpCfg object allows for the configuration of a range of sessions through the SrcPrefixAddr, GrpPrefixAddr, SrcPrefixLen, and GrpPrefixLen attributes.

Cable operators can specify configuration rules for a range of multicast sessions through the tuples of (SrcPrefixAddr, SrcPrefixLen, GrpPrefixAddr, GrpPrefixLen) attributes in an entry. The QosCfgId attribute identifies the QoS rule, the EncryptCfgId identifies the encryption rule and the PhsCfgId identifies the PHS rule for a particular entry. Even if an entry indicates a range of multicast sessions, the Encryption and PHS rules are applied on a per-session basis. Thus, when an Operator configures PHS rules or Encryption for a given GroupConfig entry, each session has those rules applied on a per session and per replication basis. Group PHS and Group Encryption rules are indicated by using a non-zero value for the PhsCfgId and EncryptCfgId, respectively.

The DocsMcastCmtsGrpQosCfgQosCtrl attribute from the CmtsGrpQosCfg object is used to determine if the traffic for a range of multicast sessions identified by an entry in the CmtsGrpCfg object will be transmitted in an "Aggregate-Session" Group Service Flow (GSF) or will be transmitted separately for each session using "Single-Session" GSFs. Even if the range of multicast sessions are transmitted on an "Aggregate-Session" GSF, the PHS and Encryption rules are always applied individually to a multicast session on a per-session DSID basis prior to being transmitted on an "Aggregate-Session" GSF.

Creation of a new instance of this object requires the following attributes to be set:

- RulePriority
- SrcPrefixAddr
- SrcPrefixLen
- GrpPrefixAddr
- GrpPrefixLen
- TosLow
- TosHigh
- TosMask

Reference: [7], CmtsGrpCfg Object section.

Table 6.100: CmtsGrpCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
CmtsGrpQosCfg	Directed aggregation to CmtsGrpQosCfg	0..*		QosConfigId
CmtsGrpPhsCfg	Directed aggregation to CmtsGrpPhsCfg	0..*	0..1	PhsConfigId
CmtsGrpEncryptCfg	Directed aggregation to CmtsGrpEncryptCfg	0..*	0..1	EncryptConfigId
Ssm	Specialization of Ssm			

6.5.7.5.5 Ssm

This configuration object is included in figure 6.10 for reference. It is defined in clause 6.5.7.7.7.

6.5.7.5.6 CmtsGrpEncryptCfg

This configuration object consists of the read-write objects of the CmtsGrpEncryptCfg object defined in Annex M of [7] and will be used with the following modification for CCAP: An enumeration of other(1) has been added to the Ctrl and Alg enumerations to allow for vendor extension. The enumeration definitions can be found in the following attributes table.

This object controls the configuration of the Security Association (SA) and the encryption algorithm used for multicast sessions.

Reference: [7], CmtsGrpEncryptCfg Object section

Table 6.101: CmtsGrpEncryptCfg Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Ctrl	Enum	No	other(1), cmts(2), mgmt(3)		mgmt
Alg	Enum	No	other(1), des56CbcMode(2), des40CbcMode(3), aes128CbcMode(4)		des56CbcMode

6.5.7.5.7 CmtsGrpPhsCfg

This configuration object consists of the read-write objects of the CmtsGrpPhsCfg object defined in Annex M of [7] and will be used without further modification for CCAP.

This object controls the configuration of DSID-indexed PHS for multicast sessions. Configuration of PHS Rules via this object are applied to individual multicast sessions even if the referenced GrpCfg object identified a GrpQosCfg instance with a QosCtrl of "aggregateSession".

This object supports the creation and deletion of instances.

Creation of multiple instances of this object require the following attributes to be set:

- PhsField
- PhsMask
- PhsSize

Reference: [7], CmtsGrpPhsCfg Object section.

6.5.7.5.8 CmtsGrpQosCfg

This configuration object consists of the read-write objects of the CmtsGrpQosCfg object defined in Annex M of [7] and will be used with the following modification for CCAP: An enumeration of other(1) has been added to the QosCtrl enumeration to allow for vendor extension. The enumeration definition can be found in table 6.102.

This object configures the QoS for Multicast sessions replicated to any Downstream Channel Set (DCS). It does not control to which particular DCSs the CCAP replicates a multicast session.

An instance of this object is called a GQC entry. A GQC entry controls how the CCAP instantiates a Group Classifier Rule (GCR) on the DCS to match packets of the multicast session. A GCR uses source and destination IP address and ToS criteria.

A GQC entry controls how and with what QoS parameters a GSF is created on a DCS. All downstream multicast packets are scheduled on a GSF. The QoS Type attribute of the GQC entry controls whether the CCAP creates one GSF for each single IP multicast session or whether the CCAP creates one GSF for the aggregate of all sessions that match the GQC criteria. The GQC instance contains a reference to a Service Class Name QoS Parameter Set template. The Service Class defines the list of QoS parameters for the GSF(s) instantiated for the GQC entry.

A CCAP identifies one Service Class as the Default Group QoS Service Class. The CCAP instantiates a Default GSF on each single-channel DCS based on the parameters of the Default Group QoS Service Class.

The set of GCRs and GSFs instantiated on a DCS control how QoS is provided to multicast packets replicated to the DCS. For each multicast packet, the CCAP classifies the packet to the highest priority matching GCR on that DCS. The GCR refers to a single GSF, which controls the scheduling of the packets on the DCS. If the multicast packet does not match any GCR on the DCS, the packet is scheduled on the Default GSF of the DCS. The CCAP replicates unclassified multicast traffic to only DCSs consisting of a single downstream channel. Thus, the Maximum Sustained Traffic Rate QoS parameter of the Default Group Service Class limits the aggregate rate of unclassified multicast traffic on each downstream channel.

The CCAP is expected to instantiate GCRs and GSFs controlled by the entries in this table only for the duration of replication of the multicast sessions matching the entry.

This object supports the creation of multiple instances.

Creation of new instances of this object require the following objects to be set:

- SvcClassName
- QosCtrl
- AggSessLimit

Reference: [7], CmtsGrpQosCfg Object section.

Table 6.102: CmtsGrpQosCfg Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
QosCtrl	Enum	Yes	other(1), singleSession(2), aggregateSession(3)		

Table 6.103: CmtsGrpQosCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
ServiceClass	Directed aggregation to ServiceClass	0..*	0..1	ServiceClassName

6.5.7.5.9 ServiceClass

This configuration object is included in figure 6.10 for reference. It is defined in clause 6.5.7.4.3.

6.5.7.5.10 DefGrpSvcClass

This configuration object is based on the DefGrpSvcClass object defined in [7] and will be used without further modification for CCAP.

This object provides the name of the Default Group Service Class. The CCAP instantiates a Default GSF with the QOS param Set indicated by this Service Class Name reference on every Downstream Channel Set to which it replicates multicast packets that are otherwise unclassified by a Group Classifier Rule.

Reference: [7], DefGrpSvcClass Object section.

Table 6.104: DefGrpSvcClass Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
ServiceClass	Directed aggregation to ServiceClass		0..1	Def

6.5.7.6 MAC Domain Configuration

The Object Model for MAC Domain configuration is shown in figure 6.11.

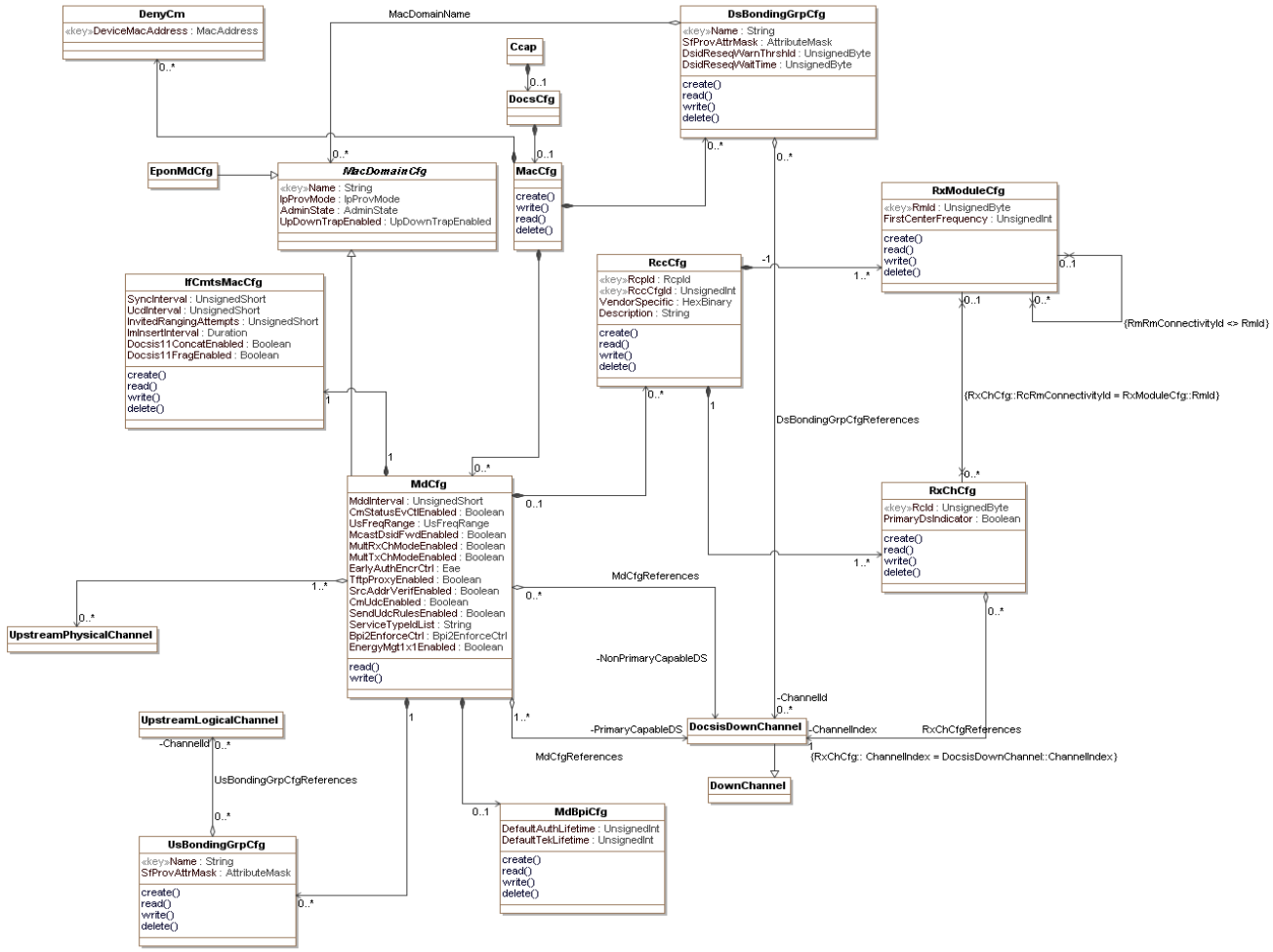


Figure 6.11: MAC Domain Configuration Objects

6.5.7.6.1 Ccap

This configuration object is included in figure 6.11 for reference. It is defined in clause 6.5.4.1.

6.5.7.6.2 DocsCfg

This configuration object is included in figure 6.11 for reference. It is defined in clause 6.5.7.1.2.

6.5.7.6.3 MacCfg

The MacCfg object is the container for DOCSIS MAC Domain configuration objects. It has the following associations:

Table 6.105: MacCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
MdCfg	Directed composition to MdCfg		0..*	
DsBondingGrpCfg	Directed composition to DsBondingGrpCfg		0..*	
DenyCm	Directed composition to DenyCm		0..*	

6.5.7.6.4 MdCfg

This object is based on the MdCfg object defined in [7]. The following modifications have been made:

- The ifIndex attribute has been removed.
- The docsIf3MdCfgDownChannelAnnex attribute has been removed.
- An enumeration of other(1) has been added to the UsFregRange, EarlyAuthEncrCtrl, and Bpi2EnforceCtrl enumerations to allow for vendor extension. The enumeration definitions can be found in table 6.106.

A MAC Domain corresponds to exactly one instance of a DocsCableMacLayer interface (ifType of 127) in the ifTable. In the configuration model, MdCfg is identified with a Name that is unique within the CCAP, defined in MacDomainCfg. For the ifTable, the CCAP implementation selects a value of the ifIndex for the DocsCableMacLayer index. The DocsCableMacLayer ifIndex is used extensively in several reporting objects as an index for several reporting objects. The CcapInterfaceIndexMapTable, defined in the CCAP MIB in Annex A, SNMP MIBs (Normative), maps a DocsCableMacLayer ifIndex to a configured MdCfg instance.

Some CCAP implementations may implement the association of non-primary capable downstream channels with MAC Domain indirectly, based on RF plant topology configuration.

Reference: [7], MdCfg Object.

Table 6.106: MdCfg Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
UsFreqRange	Enum	No	other(1), standard(2), extended(3)		standard
EarlyAuthEncrCtrl	Enum	No	other(1), disableEae(2), enableEaeRangingBasedEnforcement(3), enableEaeCapabilityBasedEnforcement(4), enableEaeTotalEnforcement(5)		enableEaeRangingBasedEnforcement
Bpi2EnforceCtrl	Enum	No	other(1), disable(2), qosCfgFileWithBpi2AndCapabPrivSupportEnabled(3), qosCfgFileWithBpi2Enabled(4), qosCfgFile(5), total(6)		qosCfgFileWithBpi2Enabled

Table 6.107: MdCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
MacDomainCfg	Specialization of MacDomainCfg			
IfCmtsMacCfg	Directed composition to IfCmtsMacCfg	1	1	
UpstreamPhysicalChannel	Directed aggregation to UpstreamPhysicalChannel	1..*	0..*	
UsBondingGrpCfg	Directed composition to UsBondingGrpCfg	1	0..*	
DocsisDownChannel	Directed aggregation to DocsisDownChannel	1..*		PrimaryCapableDs
DocsisDownChannel	Directed aggregation to DocsisDownChannel	0..*		NonPrimaryCapableDs
RccCfg	Directed composition to RccCfg	0..1	0..*	
MdBpiCfg	Directed composition to MdBpiCfg		0..1	

6.5.7.6.5 MdBpiCfg

This object is based on the DocsBpiCmtsBaseEntry table defined in RFC 3083 [12].

This optional object provides the configuration of the Baseline Privacy key lifetimes for the MAC domain. If not used, the CCAP uses the defaults defined in SysBpiCfg.

Reference: RFC 3083 [12].

6.5.7.6.6 MacDomainCfg

The MacDomainCfg abstract object contains the MAC domain attributes used by DOCSIS and EPON MAC domains.

Table 6.108: MacDomainCfg Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Name	String	Yes (Key)			
IpProvMode	Enum	Yes	other(1), ipv4Only(2), ipv6Only(3), alternate(4), dualStack(5)		
AdminState	AdminState	No			down
UpDownTrapEnabled	UpDownTrapEnabled	No			true

6.5.7.6.6.1 MacDomainCfg Object Attributes

6.5.7.6.6.1.1 Name

The name of the MacDomain.

6.5.7.6.6.1.2 IpProvMode

This attribute configures the IP provisioning mode for a MAC Domain.

6.5.7.6.6.1.3 AdminState

This attribute configures the administrative state of the MAC Domain.

6.5.7.6.6.1.4 UpDownTrapEnabled

This attribute configures whether linkUp/linkDown traps are enabled for this MAC Domain.

6.5.7.6.7 EponMdCfg

This configuration object is included in figure 6.11 for reference. It is defined in clause 6.5.11.6.

6.5.7.6.8 IfCmtsMacCfg

This object is based on the docsIfCmtsMacTable defined in RFC 4546 [20]. The following modifications have been made:

- The following attributes have been removed:
  - ifIndex
  - docsIfCmtsMacCapabilities
  - docsIfCmtsMacMaxServiceIds
  - docsIfCmtsMacStorageType
- The SynchInterval attribute (docsIfCmtsSyncInterval) data type has been shortened to UnsignedShort.
- The following attributes have been added to the IfCmtsMacCfg object, and are defined here:
  - Docsis11ConcatEnabled
  - Docsis11FragEnabled

Reference: RFC 4546 [20], docsIfCmtsMacTable

Table 6.109: IfCmtsMacCfg Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Docsis11ConcatEnabled	Boolean	No			true
Docsis11FragEnabled	Boolean	No			true

6.5.7.6.8.1 IfCmtsMacCfg Object Attributes

6.5.7.6.8.1.1 Docsis11ConcatEnabled

Enables and disables DOCSIS 1.1 concatenation.

6.5.7.6.8.1.2 Docsis11FragEnabled

Enables and disables DOCSIS 1.1 fragmentation.

6.5.7.6.9 DocsisDownChannel

This configuration object is included in figure 6.11 for reference. It is defined in clause 6.5.5.13.

6.5.7.6.10 DownChannel

This configuration object is included in figure 6.11 for reference. It is defined in clause 6.5.5.12.

6.5.7.6.11 DsBondingGrpCfg

The DsBondingGrpCfg object allows for the static creation of Downstream bonding groups. In some current DOCSIS 3.0 configurations, downstream channels are not tied directly to a specific MAC domain, while in others these downstream channels are an integral part of the MAC domain. For CCAP flexibility, the statically-configured bonding group may be optionally explicitly associated with one or multiple MAC domains.

To configure a downstream bonding group, an instance of the DsBondingGrpCfg object is created. The attributes of the DsBondingGrpCfg are shown in table 6.110 . This table has been modified from the definition in OSSIv3.0.

Table 6.110: DsBondingGrpCfg Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Name	String	Yes (Key)			
SfProvAttrMask	AttributeMask	No			bonded
DsidReseqWarnThrshld	unsignedByte	No	0..179 | 255	hundredMicroseconds	255
DsidReseqWaitTime	unsignedByte	No	1..180 | 255	hundredMicroseconds	255

Table 6.111: DsBondingGrpCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
DocsisDownChannel	Directed aggregation to DocsisDownChannel	0..*	0..*	ChannelId
MacDomainCfg	Directed aggregation to MacDomainCfg		0..*	MacDomainName

6.5.7.6.11.1 DsBondingGrpCfg Object Attributes

6.5.7.6.11.1.1 Name

The name of the downstream bonding group. This attribute is used as a key.

6.5.7.6.11.1.2 SfProvAttrMask

This attribute represents the Provisioned Attribute Mask encoding for the bonding group.

6.5.7.6.11.1.3 DsidReseqWarnThrshld

This attribute provides the DSID Resequencing Warning Threshold in hundredMicroseconds that is to be used for all DSIDs associated with this Downstream Bonding Group. The value of 255 indicates that the DSID Resequencing Warning Threshold is determined by the CMTS. The value of 0 indicates that the threshold warnings are disabled.

When the value of DsidReseqWaitTime is not equal to 0 or 255, the CCAP will ensure that the value of this object is either 255 or less than the value of DsidReseqWaitTime.

6.5.7.6.11.1.4 DsidReseqWaitTime

This attribute provides the DSID Resequencing Wait Time in hundredMicroseconds that is to be used for all DSIDs associated with this Downstream Bonding Group. The value of 255 indicates that the DSID Resequencing Wait Time is determined by the CMTS.

6.5.7.6.12 UsBondingGrpCfg

The UsBondingGrpCfg object allows for the static creation of upstream bonding groups. To configure an upstream bonding group, an instance of the UsBondingGrpCfg object is created. The attributes of the UsBondingGrpCfg are shown in table 6.112. This table has been modified from the definition in [7].

Table 6.112: UsBondingGrpCfg Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Name	String	Yes (Key)			
SfProvAttrMask	AttributeMask	No			bonded

Table 6.113: UsBondingGrpCfg Object Associations

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

6.5.7.6.12.1 UsBondingGrpCfg Object Attributes

6.5.7.6.12.1.1 Name

The name of the upstream bonding group. This attribute is used as a key.

6.5.7.6.12.1.2 SfProvAttrMask

This attribute represents the Provisioned Attribute Mask encoding for the bonding group.

6.5.7.6.13 UpstreamLogicalChannel

This configuration object is included in figure 6.11 for reference. It is defined in clause 6.5.7.8.

6.5.7.6.14 RccCfg

This object is based on the RccCfg object defined in [7] and is used with the following modification: The MdIfIndex attribute has been removed and replaced by the named association between MdCfg and RccCfg (MdCfgName).

This object creates static Receive Channel Configurations for specific downstream channel configurations, identifies the scope of the Receive Channel Configuration (RCC), and provides a top level container for the Receive Module and Receive Channel objects. The CCAP selects an instance of this object to assign to a CM when it registers.

This object supports the creation and deletion of multiple instances.

Reference: [7], RccCfg Object section.

Table 6.114: RccCfg Object Associations

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

6.5.7.6.15 RxChCfg

In the [7] configuration model, this object was indexed with the fIndex, RcpId, RccCfgId, and RcId. For CCAP, this object will use RcId as the index. The following attributes have been removed and are replaced by the object's associations:

- IfIndex
- RcpId
- RccCfgId
- ChIfIndex

The Receive Channel Configuration object permits an operator to configure how CMs registered with certain Receive Channel Profiles will configure the Receive Channels within their profile.

When a CM registers with a Receive Channel Profile (RCP) for which all Receive Channel Indices (RcIds) are configured in the Receive Module object and all Receive Channels are configured within this object, the CCAP should use the configuration within these objects to set the Receive Channel Configuration returned to the CM in a REG-RSP message.

The CCAP may require configuration of all pertinent Receive Module and Receive Channel instances in order to register a CM that reports a Receive Channel Profile (RCP), including any standard Receive Channel Profiles.

If the CM reports multiple RCPs, and Receive Module and Receive Channel objects have instances for more than one RCP, the particular RCP selected by the CCAP is not specified. A CCAP is not restricted to assigning Receive Modules based only on the contents of this object.

This object supports the creation and deletion of multiple instances.

Creation of a new instance of this object requires the reference of a valid RccCfg instance and the ChIfIndex attribute to be set.

Reference: [7], RxChCfg Object section.

Table 6.115: RxChCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
RxModuleCfg	Association with RxModuleCfg	0..*	0..1	
DocsisDownChannel	Directed aggregation to DocsisDownChannel			ChannelIndex

6.5.7.6.16 RxModuleCfg

In the DOCSIS 3.0 configuration mode, this object was indexed with the MacDomainIfIndex, RcpId, RccCfgId, and RmId. For CCAP, this object will use RmId as the index. The following attributes have been removed and are replaced by the objects associations:

- IfIndex
- RcpId
- RccCfgId

The rest of the configuration object is the same as [7].

The Receive Module Configuration object permits an operator to configure how CMs with certain RCPs will configure the Receive Modules within their profile upon CM registration.

When a CM registers with an RCP for which all Receive Module Indices (RmIds) are configured in this object and all Receive Channels are configured within the Receive Channel (RxCh) object, the CCAP should use the configuration within these objects to set the Receive Channel Configuration assigned to the CM in a REG-RSP message.

The CCAP may require configuration of all pertinent Receive Module and Receive Channel instances in order to register a CM that reports a Receive Channel Profile.

If the CM reports multiple RCPs, and Receive Module and Receive Channel objects have instances for more than one RCP reported by the CM, the particular RCP selected by the CCAP is not specified. A CCAP is not restricted to assigning Receive Modules based only on the contents of this object.

This object supports the creation and deletion of multiple instances.

Creation of a new instance of this object requires the reference of a valid RccCfg instance.

Reference: [7], RxModuleCfg Object section.

Table 6.116: RxModuleCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
RxChCfg	Association with RxChCfg	0..1	0..*	
RxModuleCfg	Association with RxModuleCfg	0..1	0..*	

The CCAP shall reject the configuration of an instance of RxModuleCfg that is associated with itself. The CCAP shall reject the configuration of an instance of RxChCfg instances with circular references.

6.5.7.6.17 DenyCm

This configuration object allows an operator to create a list of CM MAC addresses that are not allowed to register.

Table 6.117: DenyCm Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
DeviceMacAddress	MacAddress	Yes (Key)			

6.5.7.6.17.1 DenyCm Object Attributes

6.5.7.6.17.1.1 DeviceMacAddress

The MAC address of the CM that will be added to the deny list. This attribute is used as a key.

6.5.7.7 DOCSIS® Multicast Authorization Configuration

This group of configuration elements allows for the configuration of DOCSIS Multicast Authorization. They are based on the configuration elements from [7]. The configuration specific Object Model is shown in figure 6.12.

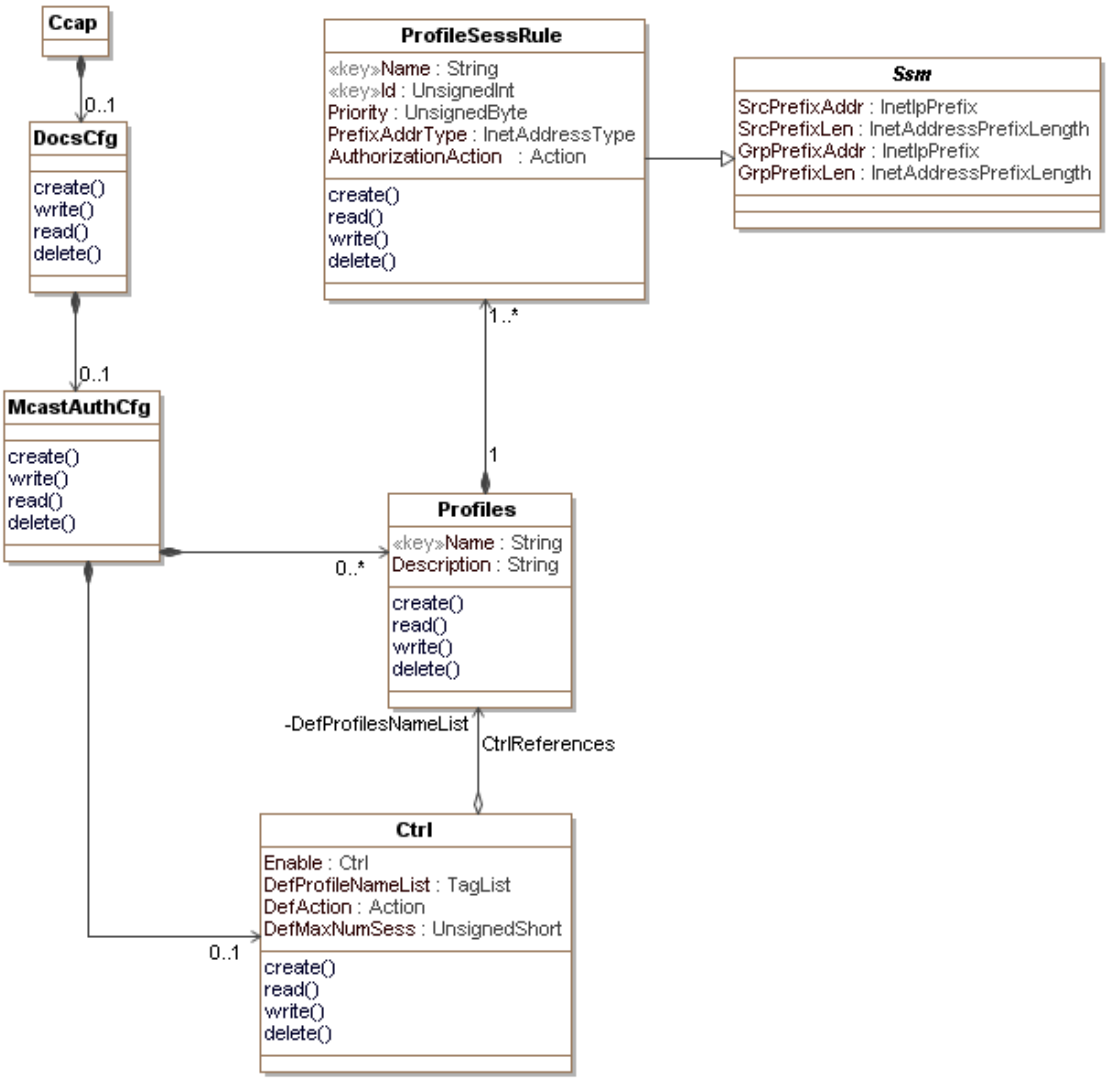


Figure 6.12: DOCSIS® Multicast Authorization Configuration Objects

6.5.7.7.1 Ccap

This configuration object is included in figure 6.12 for reference. It is defined in clause 6.5.4.1.

6.5.7.7.2 DocsCfg

This configuration object is included in figure 6.12 for reference. It is defined in clause 6.5.7.1.2.

6.5.7.7.3 McastAuthCfg

The McastAuthCfg object is the container for DOCSIS Multicast Authorization configuration objects. It has the following associations:

Table 6.118: McastAuthCfg Object Associations

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

6.5.7.7.4 Profiles

This configuration object is based on the read-write objects of the Profiles object defined in Annex M of [7] and will be used without further modification for CCAP.

This object contains the description of the Multicast Authorization profiles for administrative purposes.

This object supports the creation and deletion of multiple instances.

Creation of a new instance of this object requires the Description attribute to be set.

Reference: [7], Profiles Object section.

Table 6.119: Profiles Object Associations

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

6.5.7.7.5 Ctrl

This configuration object is based on the Ctrl object defined in Annex M of [7] and will be used with the following modification for CCAP: An enumeration of other(1) has been added to the Enable and DefAction enumerations to allow for vendor extension. The enumeration definitions can be found in the following attributes table.

This object defines the CCAP global behavior for Multicast Authorization. Some parameters are included as part of the CM configuration process. In absence of those parameters, default values defined by attributes of this object are used.

Reference: [7], Ctrl Object section.

Table 6.120: Ctrl Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units [7]	Default Value
Enable	Enum	No	other(1), enable(2), disable(3)		disable
DefAction	Enum	No	other(1), accept(2), deny(3)		deny

Table 6.121: Ctrl Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
Profiles	Directed aggregation to Profiles			DefProfilesNameList

6.5.7.7.6 ProfileSessRule

This configuration object is based on the read-write objects of the ProfileSessRule object defined in Annex M of [7] and will be used with the following modifications for CCAP:

- The PrefixAddrType has been removed.
- The Action attribute has been renamed AuthorizationAction.
- An enumeration of other(1) has been added to the AuthorizationAction enumeration to allow for vendor extension. The enumeration definitions can be found in able 6.122.

- The following attributes have been moved into the abstract class Ssm:
  - SrcPrefixAddr
  - SrcPrefixLen
  - GrpPrefixAddr
  - GrpPrefixLen

This object defines Operator configured profiles to be matched during the authorization process.

This object supports the creation and deletion of multiple instances.

Creation of a new instance of this object requires the following attributes to be set:

- SrcPrefixAddr
- SrcPrefixLen
- GrpPrefixAddr
- GrpPrefixLen

Reference: [7], ProfileSessRule Object section.

Table 6.122: ProfileSessRule Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
AuthorizationAction	Enum	No	other(1), accept(2), deny(3)		deny

6.5.7.7.7 Ssm

This abstract object holds the shared session attributes of the ProfileSessRule and the CmtsGrpCfg objects. It contains the following attributes, which are based on the read-write attributes of the ProfileSessRule object defined in Annex M of [OSSIv3.0]:

- SrcPrefixAddr
- SrcPrefixLen
- GrpPrefixAddr
- GrpPrefixLen

6.5.7.8 DOCSIS® Interface Configuration

The DOCSIS Interface configuration objects are shown in figure 6.13.

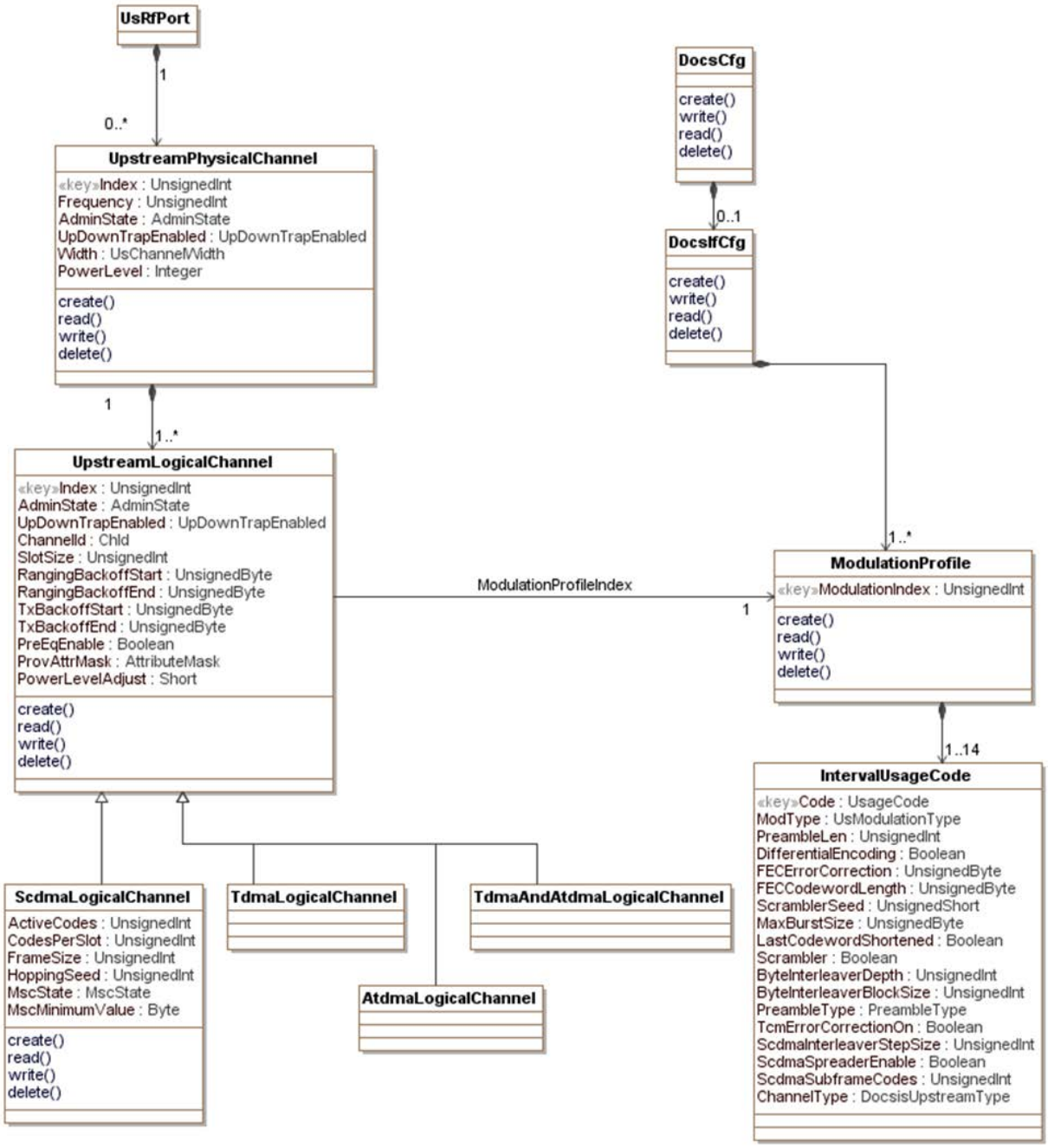


Figure 6.13: DOCSIS® Interface Configuration Objects

6.5.7.8.1 DocsCfg

This configuration object is included in figure 6.13 for reference. It is defined in clause 6.5.7.1.2.

6.5.7.8.2 DocsIfCfg

The DocsIfCfg object is the container for the DOCSIS interface configuration objects. It has the following associations:

Table 6.123: DocsIfCfg Object Associations

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

6.5.7.8.3 ModulationProfile

This object allows a modulation profile to be associated to an upstream logical channel. It has a single attribute, ModulationIndex, which is based on the Index attribute defined in docsIfCmtsModulationTable defined in [20].

Reference: [20], docsIfCmtsModulationTable.

Table 6.124: ModulationProfile Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
IntervalUsageCode	Directed composition to IntervalUsageCode		1..14	

6.5.7.8.4 IntervalUsageCode

This object allows a list of interval usage codes to be associated with a single modulation profile. It is based on the docsIfCmtsModulationTable defined in RFC 4546 [20] and will be used with the following modifications for CCAP. The following attributes have been removed:

- ModulationIndex (included in the ModulationProfile object)
- StorageType
- Control
- GuardTimeSize

The IntervalUsageCode attribute has been renamed Code.

The ModType, PreambleType and ChannelType attributes have had the unknown enumerations removed and a new enumeration, other(1), added to allow for vendor extension. The enumeration definitions can be found in the following attributes table.

Reference: RFC 4546 [20], docsIfCmtsModulationTable.

Table 6.125: IntervalUsageCode Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
ModType	Enum	No	other(1), qpsk(2), qam8(3), qam16(4), qam32(5), qam64(6), qam128(7)		qpsk
PreambleType	Enum	Yes	other(1), qpsk0(2), qpsk1(3)		
ChannelType	Enum	Yes	other(1), tdma(2), atdma(3), scdma(4), tdmaAtdma(5)		

6.5.7.8.5 UsRfPort

This configuration object is included in figure 6.13 for reference. It is defined in clause 6.5.5.19.

6.5.7.8.6 UpstreamPhysicalChannel

The UpstreamPhysicalChannel object represents DOCSIS operation on a single upstream center frequency at a particular channel width.

Since CCAP is expected to operate with only DOCSIS 2.0 or later upstream channels, at least one UpstreamLogicalChannel object (ifType 205) is needed to be instantiated to operate within an UpstreamPhysicalChannel.

This object differs from the same object in DOCSIS in that the desired input power is now set at the UpstreamPhysicalChannel and not on a per-UpstreamLogicalChannel instance. If the target receive power level for an individual logical channel under a physical channel is desired to be different than the target power level for the physical channel, this can be configured using the PowerLevelAdjust attribute of the UpstreamLogicalChannel object.

Table 6.126: UpstreamPhysicalChannel Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes (Key)	1..*		
Frequency	UnsignedInt	Yes	5 000 000..85 000 000	Hertz	
Width	Enum	Yes	other(1), 200 000(2), 400 000(3), 800 000(4) 1 600 000(5), 3 200 000(6), 6 400 000(7)	Hertz	
AdminState	AdminState	No			down
UpDownTrapEnabled	UpDownTrapEnabled	No			true
PowerLevel	Integer	Yes		TenthdBmV	

An UpstreamPhysicalChannel is contained by a single UsRfPort. It contains one or more UpstreamLogicalChannel objects. It is referenced by a single MacDomain.

Table 6.127: UpstreamPhysicalChannel Object Associations

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

6.5.7.8.6.1 UpstreamPhysicalChannel Requirements

The CCAP shall reject activation of a set of configuration objects that would cause an overlap of RF channel frequency on any single upstream RF port.

6.5.7.8.6.2 UpstreamPhysicalChannel Object Attributes

6.5.7.8.6.2.1 Index

This attribute uniquely identifies an UpstreamPhysicalChannel on its UsRfPort. Its value is between one and the maximum number of UpstreamPhysicalChannels supported on the UsRfPort, inclusive.

6.5.7.8.6.2.2 Frequency

This attribute configures the center frequency of the UpstreamPhysicalChannel, in Hertz. As of DOCSIS 3.0, the minimum permitted value is the center frequency such that the lower channel edge is 5 000 000 Hz and the maximum permitted value is the center frequency at which the upper channel edge is 85 000 000 Hz. This attribute corresponds to the docsIfUpChannelFrequency object of DOCS-IF-MIB RFC 4546 [20]. The CCAP shall reject the configuration of an UpstreamPhysicalChannel instance that overlaps in frequency with another UpstreamPhysicalChannel instance on the same upstream RF port.

6.5.7.8.6.2.3 Width

This attribute configures the width of the UpstreamPhysicalChannel, in Hertz. While the only permitted values as of DOCSIS 3.0 are 1 600 000, 3 200 000, and 6 400 000, the present document also includes widths of 200 000, 400 000, and 800 000 for backward compatibility. This attribute corresponds to the docsIfUpChannelFrequency object of DOCS-IF-MIB RFC 4546 [20].

The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.7.8.6.2.4 AdminState

This attribute configures the administrative state of this instance.

6.5.7.8.6.2.5 UpDownTrapEnabled

This attribute configures whether linkUp/linkDown traps are enabled for this channel.

6.5.7.8.6.2.6 PowerLevel

This attribute configures the desired input power level, in TenthdBmV, common to all upstream logical channels associated with this physical channel instance. The power level for an individual logical channel can deviate from the common power level through the configuration of the PowerLevelAdjust attribute of the UpstreamLogicalChannel object.

6.5.7.8.7 UpstreamLogicalChannel

The UpstreamLogicalChannel object represents scheduled intervals of time on a single UpstreamPhysicalChannel. An UpstreamLogicalChannel is either SCDMA, TDMA, ATDMA, or both TDMA and ATDMA. Each UpstreamLogicalChannel is identified with a DOCSIS upstream channel ID. The MAP management messages transmitted downstream by the CCAP schedule intervals of time for each DOCSIS upstream channel ID. In the SNMP MIB, an UpstreamLogicalChannel is an interface with ifType UpstreamLogicalChannel (205).

Table 6.128: UpstreamLogicalChannel Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes (Key)			
AdminState	AdminState	No			down
UpDownTrapEnabled	UpDownTrapEnabled	No			false
ChannelId	UnsignedInt	No			0
SlotSize	UnsignedInt	Yes		ticks	
RangingBackoffStart	UnsignedByte	Yes	0..16	power of 2	
RangingBackoffEnd	UnsignedByte	Yes	0..16	power of 2	
TxBackoffStart	UnsignedByte	Yes	0..16		
TxBackoffEnd	UnsignedByte	Yes	0..16		
PreEqEnable	Boolean	Yes			
ProvAttrMask	AttributeMask	Yes			
PowerLevelAdjust	Short	No		TenthdB	0

This object differs from the same object in DOCSIS in that the desired common input power is now set at the Upstream Physical Channel and power level adjustments can only be configured on a per UpstreamLogicalChannel basis.

Table 6.129: UpstreamLogicalChannel Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
ModulationProfile	Directed association to ModulationProfile		1	ModulationProfileIndex

6.5.7.8.7.1 UpstreamLogicalChannel Object Attributes

6.5.7.8.7.1.1 Index

This key attribute uniquely identifies an UpstreamLogicalChannel operating on the center frequency and width of a single UpstreamPhysicalChannel. This index is in the range between one and the maximum number of UpstreamLogicalChannel objects supported by the CCAP on an UpstreamPhysicalChannel.

6.5.7.8.7.1.2 AdminState

This attribute stores the administrative state of the upstream logical channel.

6.5.7.8.7.1.3 UpDownTrapEnabled

This attribute configures whether linkUp/linkDown traps are enabled for this channel.

6.5.7.8.7.1.4 ChannelId

This attribute permits an operator to optionally configure the upstream channel ID signaled in the DOCSIS protocol for the UpstreamLogicalChannel. By default, the CCAP will automatically assign the DocsisUpChannelId. An operator can create or update this attribute with a value to force the CCAP to use the configured DOCSIS channel ID. A unique configured value exists within the MacDomain to which the UpstreamPhysicalChannel containing this UpstreamLogicalChannel is associated. A value of 0 means that the CCAP should automatically assign the ChannelId.

6.5.7.8.7.1.5 SlotSize

This attribute configures the number of 6,25 microsecond ticks in each upstream mini-slot for the UpstreamLogicalChannel. This attribute may have different values for the different UpstreamLogicalChannel objects on the same UpstreamPhysicalChannel. This attribute is applicable to TDMA and ATDMA channel types only; its value is read and written as zero for SCDMA type channels.

6.5.7.8.7.1.6 RangingBackoffStart

This attribute is the initial random back-off window to use when retrying Ranging Requests. It is expressed as a power of 2. A configured value of 16 indicates that a proprietary adaptive retry mechanism is to be used.

6.5.7.8.7.1.7 RangingBackoffEnd

This attribute is the final random back-off window to use when retrying Ranging Requests. It is expressed as a power of 2. A configured value of 16 indicates that a proprietary adaptive retry mechanism is to be used.

6.5.7.8.7.1.8 TxBackoffStart

The initial random back-off window to use when retrying transmissions. Expressed as a power of 2. A configured value of 16 indicates that a proprietary adaptive retry mechanism is to be used. See the associated conformance object for write conditions and limitations.

6.5.7.8.7.1.9 TxBackoffEnd

The final random back-off window to use when retrying transmissions. Expressed as a power of 2. A configured value of 16 indicates that a proprietary adaptive retry mechanism is to be used. See the associated conformance object for write conditions and limitations.

6.5.7.8.7.1.10 PreEqEnable

This attribute enables pre-equalization on the UpstreamLogicalChannel when its value is true, or disables pre-equalization when its value is false.

6.5.7.8.7.1.11 ProvAttrMask

This attribute configures the 32-bit Provisioned Attribute Mask for the UpstreamLogicalChannel. This is used by a CCAP to control how upstream service flows are assigned to the UpstreamLogicalChannel.

6.5.7.8.7.1.12 PowerLevelAdjust

This attribute configures the adjustment from the common power level configured for the physical Upstream (US) channel; it is expressed in TenthdB. The sum of the UpstreamPhysicalChannel PowerLevel and UpstreamLogicalChannel PowerLevelAdjust determines the expected input power level for the logical channel. If the CCAP does not support the ability to set the PowerLevelAdjust attribute to a non-zero value, the CCAP may log an error upon execution of an XML configuration file that contains a negative attribute value.

6.5.7.8.8 ScdmaLogicalChannel

This configuration object is constructed from the SCDMA fields of the docsIfUpstreamChannelTable defined in RFC 4546 [20] and the DOCS-IFEXT2-MIB defined in Annex H of [7], and these attributes are used with the following modification for CCAP: a value of "other" has been added to the MscState attribute's enumeration to allow for vendor extension. The enumeration definition can be found in the following attributes table.

The Scdma object is an optional grouping of additional parameters to an UpstreamLogicalChannel that is defined only for UpstreamLogicalChannel objects that reference an SCDMA modulation profile.

References: RFC 4546 [20], docsIfUpstreamChannelTable; [7], Annex H.

Table 6.130: ScdmaLogicalChannel Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
MscState	Enum	No	other(1), channelEnabled(2), channelDisabled(3), dormant(4)		channelDisabled

Table 6.131: ScdmaLogicalChannel Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Units	Default Value
UpstreamLogicalChannel	Specialization of UpstreamLogicalChannel				

6.5.7.8.9 TdmaLogicalChannel

This configuration object is a specialization of the docsIfUpstreamChannelTable defined in RFC 4546 [20] for TDMA logical channels.

References: RFC 4546 [20], docsIfUpstreamChannelTable; [7], Annex H.

Table 6.132: TdmaLogicalChannel Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Units	Default Value
UpstreamLogicalChannel	Specialization of UpstreamLogicalChannel				

6.5.7.8.10 AtdmaLogicalChannel

This configuration object is a specialization of the docsIfUpstreamChannelTable defined in RFC 4546 [20] for ATDMA logical channels.

References: RFC 4546 [20], docsIfUpstreamChannelTable; [7], Annex H.

Table 6.133: AtdmaLogicalChannel Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Units	Default Value
UpstreamLogicalChannel	Specialization of UpstreamLogicalChannel				

6.5.7.8.11 TdmaAndAtdmaLogicalChannel

This configuration object is a specialization of the docsIfUpstreamChannelTable defined in RFC 4546 [20] for mixed TDMA/ATDMA logical channels.

References: RFC 4546 [20], docsIfUpstreamChannelTable; [7], Annex H.

Table 6.134: TdmaAndAtdmaLogicalChannel Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Units	Default Value
Scdma	Specialization of UpstreamLogicalChannel				

6.5.7.9 DSG Configuration

The CCAP incorporates the DSG Agent, which is defined as the implementation of the DSG protocol within the CCAP. The DSG Agent creates the DSG Tunnel, places content from the DSG Server into the DSG Tunnel, and sends the DSG Tunnel to the DSG Client.

For CCAP, the DSG Agent configuration object model changes slightly for several tables. The object model for the CCAP is shown in the following class diagram.

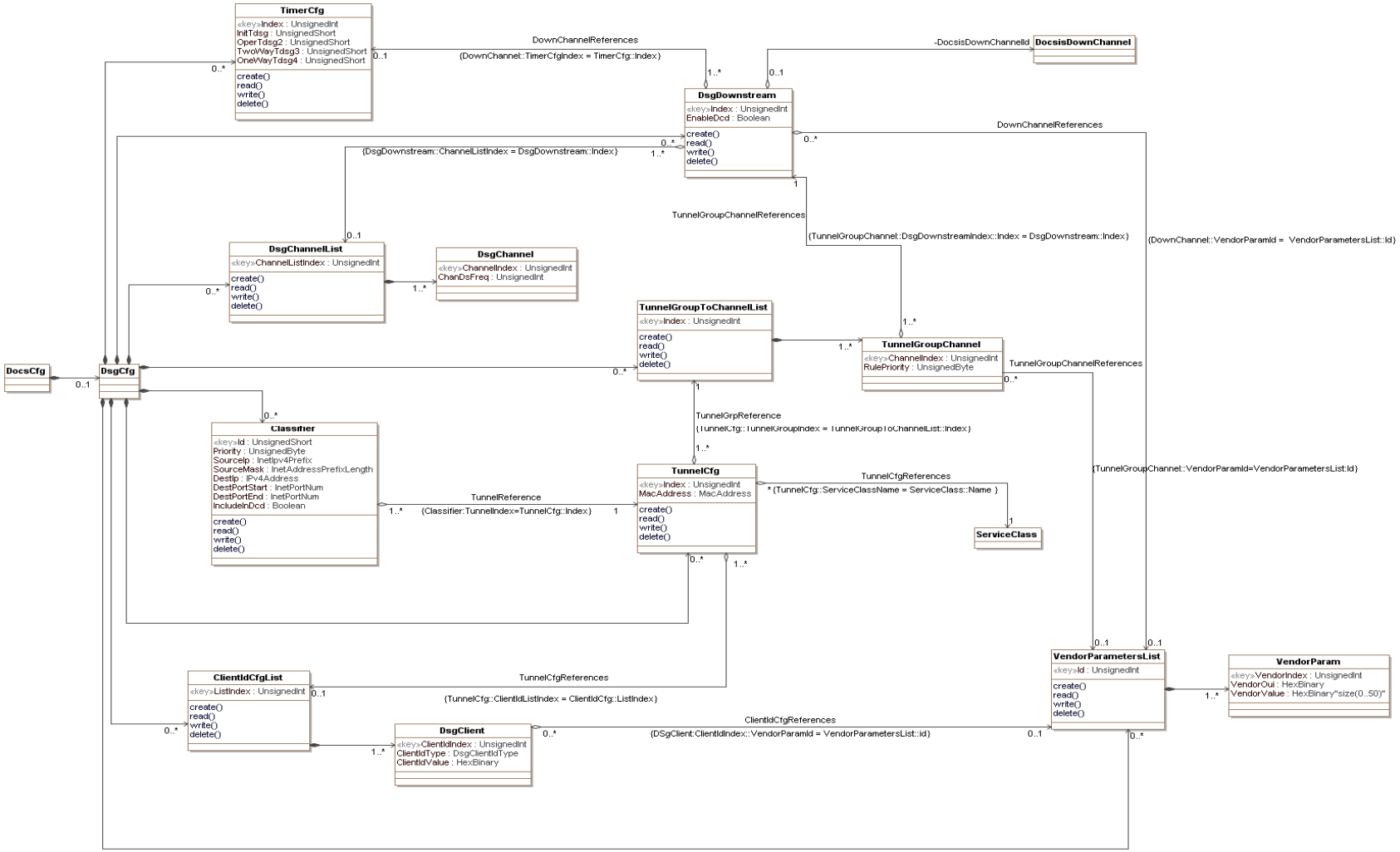


Figure 6.14: DSG Configuration Objects

6.5.7.9.1 DocsCfg

This configuration object is included in figure 6.14 for reference. It is defined in clause 6.5.7.1.2.

6.5.7.9.2 DsgCfg

The DsgCfg object is the container for DSG configuration objects. It has the following associations:

Table 6.135: DsgCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
TimerCfg	Directed composition to TimerCfg		0..*	
DsgDownstream	Directed composition to DsgDownstream		0..*	
DsgChannelList	Directed composition to DsgChannelList		0..*	
TunnelGroupToChannelList	Directed composition to TunnelGroupToChannelList		0..*	
Classifier	Directed composition to Classifier		0..*	
TunnelCfg	Directed composition to TunnelCfg		0..*	
ClientIdCfgList	Directed composition to ClientIdCfgList		0..*	
VendorParametersList	Directed composition to VendorParametersList		0..*	

6.5.7.9.3 TimerCfg

This configuration object is based on the dsgIfTimerTable defined in [i.3] and will be used with modifications for CCAP.

The DSG Timer Table contains timers that are sent to the DSG client(s) via the DCD message.

Reference: [i.3], DOCSIS Set-top Gateway Agent MIB Definition section.

Table 6.136: TimerCfg Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes (Key)			
InitTdsg	UnsignedShort	No	1..65 535	Seconds	2
OperTdsg2	UnsignedShort	No	1..65 535	Seconds	600
TwoWayTdsg3	UnsignedShort	No		Seconds	300
OneWayTdsg4	UnsignedShort	No		Seconds	1 800

6.5.7.9.3.1 TimerCfg Attributes

6.5.7.9.3.1.1 Index

The index for this object.

6.5.7.9.3.1.2 InitTdsg

Initialization Timeout. This is the timeout period in seconds for the DSG packets during initialization of the DSG client. The default value is 2 seconds.

6.5.7.9.3.1.3 OperTdsg2

Operational Timeout. This is the timeout period in seconds for the DSG packets during normal operation of the DSG client. Default value is 600 seconds.

6.5.7.9.3.1.4 TwoWayTdsg3

Two-way retry timer. This is the retry timer that determines when the DSG client attempts to reconnect with the DSG Agent and established two-way connectivity. Default value is 300 seconds. The value 0 indicates that the client will continuously retry two-way operation.

6.5.7.9.3.1.5 OneWayTdsg4

One-way retry timer. This is the retry timer that determines when the client attempts to rescan for a DOCSIS downstream channel that contains DSG packets after a TimerTdsg1 or TimerTdsg2 timeout. Default value is 1 800 seconds. Setting the value to 0 indicates that the client will immediately begin scanning upon TimerTdsg1 or TimerTdsg2 timeout.

6.5.7.9.4 DsgDownstream

The DsgDownstream object represents an individual downstream channel for DSG configuration purposes. It has been modified from the DSG Specification definitions.

Table 6.137: DsgDownstream Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes (Key)			
EnableDcd	Boolean	Yes			

The DsgDownstream object has the following associations.

Table 6.138: DsgDownstream Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
TimerCfg	Directed aggregation to TimerCfg	1..*	0..1	
DsgChannelList	Directed aggregation to DsgChannelList	1..*	0..1	
DocsisDownChannel	Directed aggregation to DocsisDownChannel	0..1		DocsisDownChannelId
VendorParametersList	Directed aggregation to VendorParametersList	0..*	0..1	

6.5.7.9.4.1 DsgDownstream Attributes

6.5.7.9.4.1.1 Index

This is the key for an instance of this object.

6.5.7.9.4.1.2 EnableDcd

This attribute is used to enable or disable DCD messages to be sent on this downstream channel. The value is always true for those downstreams that contain DSG tunnels.

6.5.7.9.5 DocsisDownChannel

This configuration object is included in figure 6.14 for reference. It is defined in clause 6.5.5.13.

6.5.7.9.6 DsgChannelList

This configuration object is based on the dsgIfChannelListTable defined in [i.3] and will be used with modifications for CCAP.

The DsgChannelList object allows for configuration of a list of one or multiple downstream frequencies that are carrying DSG tunnel(s). This configuration object has been modified from the DSG Specification definitions.

Reference: [i.3], DOCSIS Set-top Gateway Agent MIB Definition section.

Table 6.139: DsgChannelList Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
ChanListIndex	UnsignedInt	Yes (Key)			

Table 6.140: DsgChannelList Object Associations

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

6.5.7.9.6.1 DsgChannelList Attributes

6.5.7.9.6.1.1 ChanListIndex

The index of the down channel list.

6.5.7.9.7 DsgChannel

This configuration object allows for one or more downstream frequencies that are carrying DSG tunnel(s) to be associated with a DsgChannelList.

Table 6.141: DsgChannel Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
ChannelIndex	UnsignedInt	Yes (Key)			
ChanDsFreq	UnsignedInt	No	0..1 000 000 000	Hz	0

6.5.7.9.7.1 DsgChannel Attributes

6.5.7.9.7.1.1 ChannelIndex

The index of the channel.

6.5.7.9.7.1.2 ChanDsFreq

The ChanDsFreq attribute represent a frequency of a downstream channel carrying DSG information. Frequency is a multiple of 62 500 Hz, per [i.3].

6.5.7.9.8 TunnelGroupToChannelList

This configuration object is based on the dsgIfTunnelGrpToChannelTable defined in [i.3] and will be used with modifications for CCAP.

The TunnelGroupToChannelList object permits association of a group of DsgDownstream objects to one or more tunnels. This configuration object has been modified from the DSG Specification definitions.

Reference: [i.3], DOCSIS Set-top Gateway Agent MIB Definition section.

Table 6.142: TunnelGroupToChannelList Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes (Key)			

The TunnelGrpToChannel object has the following associations.

Table 6.143: TunnelGrpToChannel Object Associations

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

6.5.7.9.8.1 TunnelGroupToChannelList Attributes

6.5.7.9.8.1.1 Index

This attribute is the key for this object and allows a link to an instance of a TunnelCfg object be configured.

6.5.7.9.9 TunnelGroupChannel

The TunnelGroupChannel object allows DsgDownstream objects to be associated with this group.

Table 6.144: TunnelGroupChannel Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
ChannelIndex	UnsignedInt	Yes (Key)			
RulePriority	UnsignedByte	No	0..255		0

The TunnelGroupChannel object has the following associations.

Table 6.145: TunnelGroupChannel Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
DsgDownstream	Directed aggregation to DsgDownstream	1..*	1	
VendorParametersList	Directed association to VendorParametersList	0..*	0..1	

6.5.7.9.9.1 TunnelGroupChannel Attributes

6.5.7.9.9.1.1 ChannelIndex

This attribute configures the linkage of a specific DsgDownstream instance to the TunnelCfg instance associated with the group.

6.5.7.9.9.1.2 RulePriority

The DSG rule priority determines the order in which a channel should be applied by the DSG client. The default value is 0, which is the lowest priority.

6.5.7.9.10 Classifier

This configuration object is based on the dsgIfClassifierTable defined in [i.3] and will be used with modifications for CCAP.

Reference: [i.3], DOCSIS Set-top Gateway Agent MIB Definition section.

Table 6.146: Classifier Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Id	UnsignedShort	Yes (Key)			
Priority	UnsignedByte	No			0
SourceIp	Ipv4Prefix	Yes			
SourceMask	InetAddressPrefixLength	No			32
DestIp	Ipv4Address	Yes			
DestPortStart	InetPortNum	No			0
DestPortEnd	InetPortNum	No			65 535
IncludeInDcd	Boolean	No			true

Table 6.147: Classifier Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
TunnelCfg	Directed aggregation to TunnelCfg	1..*	1	

6.5.7.9.10.1 Classifier Attributes

6.5.7.9.10.1.1 Id

This attribute configures the linkage between the DSG tunnel for which this classifier will apply.

6.5.7.9.10.1.2 Priority

This attribute is used to configure the DSG rule priority that determines the order in which a channel and its associated UCIDs should be applied by the DSG client. The default value is 0, which is the lowest priority.

6.5.7.9.10.1.3 SourceIp

This attribute configures the source IP address for the DSG tunnel. Currently, the CCAP only supports IPv4 addresses for DSG tunnels, per [i.3].

6.5.7.9.10.1.4 SourceMask

This attribute configures the source IP address mask for the DSG tunnel.

6.5.7.9.10.1.5 DestIp

This attribute configures the destination IP address for the DSG tunnel. Currently, the CCAP only supports IPv4 addresses for DSG tunnels, per [i.3].

6.5.7.9.10.1.6 DestPortStart

This attribute configures the inclusive lower bound of the transport-layer source port range that is to be matched.

6.5.7.9.10.1.7 DestPortEnd

This attribute configures the inclusive higher bound of the transport-layer source port range that is to be matched.

6.5.7.9.10.1.8 IncludeInDcd

Indicates whether or not this DSG classifier will be sent in DCD messages for use as a Layer-3 and Layer-4 packet filter by the DSG eCM.

6.5.7.9.11 TunnelCfg

A TunnelCfg object allows the operator to configure DSG tunnels. Each DSG Tunnel represents a stream of packets delivered to a DSG Client in a set-top device and is configured with a single destination MAC address.

This configuration object is based on the dsgIfTunnelTable defined in [i.3] and is used with modifications.

Reference: [i.3], DOCSIS Set-top Gateway Agent MIB Definition section.

Table 6.148: TunnelCfg Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes (Key)			
MacAddress	MacAddress	Yes			

The TunnelCfg object has the following associations.

Table 6.149: TunnelCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
TunnelGroupToChannelList	Directed association to TunnelGroupToChannelList	1..*	1	
ClientIdCfgList	Directed aggregation to ClientIdCfgList	1..*	0..1	
ServiceClass	Directed aggregation to ServiceClass	*	1	

6.5.7.9.11.1 TunnelCfg Attributes

6.5.7.9.11.1.1 Index

This attribute is the index for a tunnel that could be associated to one or more downstream channels that carry DSG tunnels.

6.5.7.9.11.2 MacAddress

This attribute configures the DSG tunnel destination MAC address.

6.5.7.9.12 ServiceClass

This configuration object is included in figure 6.14 for reference. It is defined in clause 6.5.7.4.3.

6.5.7.9.13 ClientIdCfgList

This configuration object is based on the dsgIfClientIdTable defined in [i.3] and will be used with modifications for CCAP.

The Client Identification object contains a list of client identification types and values. Each entry in the list also contains the vendor-specific parameter identification. There could be multiple client ids associated to a tunnel, grouped by the ListIndex.

Reference: [i.3], DOCSIS Set-top Gateway Agent MIB Definition section.

Table 6.150: ClientIdCfgList Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
ListIndex	UnsignedInt	Yes (Key)			

The ClientIdCfgList object has the following associations.

Table 6.151: ClientIdCfgList Object Associations

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

6.5.7.9.13.1 ClientIdCfgList Attributes

6.5.7.9.13.1.1 ListIndex

This attribute is the key for the ClientIdCfgList object and provides the unique identifier for each client list.

6.5.7.9.14 DsgClient

The DsgClient object represents a list entry in the ClientIdCfgList object.

Reference: [i.3], DOCSIS Set-top Gateway Agent MIB Definition section.

Table 6.152: DsgClient Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
ClientIdIndex	UnsignedInt	Yes (Key)			
ClientIdType	Enum	No	other(1), broadcast(2), macAddress(3), caSystemId(4), applicationId(5)		broadcast
ClientIdValue	HexBinary	No	size(6)		'000000000000'h

The DsgClient object has the following associations.

Table 6.153: DsgClient Object Associations

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

6.5.7.9.14.1 DsgClient Attributes

6.5.7.9.141.1 ClientIdIndex

This attribute is the key and provides the unique identifier of each DsgClient object in this instance of DsgClient.

6.5.7.9.14.1.2 ClientIdType

The Client Identification type. A DSG client ID of broadcast(2) is received by all DSG clients. A DSG client ID of macAddress(3) is received by the DSG client that has been assigned with this MAC address where the first 3 bytes is the Organization Unique Identifier (OUI). A DSG client ID of caSystemId(4) is received by the DSG client that has been assigned a CA_system_ID. A DSG client ID of applicationId(5) is received by the DSG client that has been assigned an application ID. The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.7.9.14.1.3 ClientIdValue

The Client Identification Value. The content depends on the value of the dsgIfClientIdType. For dsgIfClientIdType broadcast(1), this object will have a 16-bit value whether or not it is a length 0 or length 2 broadcast ID. If the value is 0, then the encoded Type Length Value Attribute (TLV) in the DCD would be the original, zero length, broadcast ID. If the value is specified in table 5-2 of [i.3], then the TLV in the DCD would be a length 2 broadcast ID followed by the value.

For ClientIdType macAddress(2), this object is a well-known MAC address.

For ClientIdType caSystemId(3), this object is a CA System ID.

For ClientIdType applicationId(4), this object is an application ID.

Client IDs representing types broadcast(1), caSystemId(3) or applicationId(4) are encoded in DCD messages as unsigned integers and configured in this object as 6 octet string with the 2 LSB for the client ID value; e.g. an applicationId 2048 (0x0800) is encoded as '000000000800'h.

6.5.7.9.15 VendorParametersList

This configuration object is based on the dsgIfVendorParamTable defined in [i.3] and is used with the following modifications for CCAP: a VendorParam object has been created to allow a list of vendor parameters to be associated with this object.

The VendorParametersList object allows vendors to send specific parameters to the DSG clients within a DSG rule or within the DSG Configuration block in a DCD message.

Reference: [i.3], DOCSIS Set-top Gateway Agent MIB Definition section.

Table 6.154: VendorParametersList Object Associations

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

6.5.7.9.16 VendorParam

This configuration object is based on the dsgIfVendorParamTable defined in [i.3] and holds the attributes that define each vendor parameter.

Reference: [i.3], DOCSIS Set-top Gateway Agent MIB Definition section.

6.5.7.10 IPCablecom Configuration Objects

This section defines the configuration objects needed for configuring IPCablecom and IPCablecom Multimedia (PCMM) services on the CCAP.

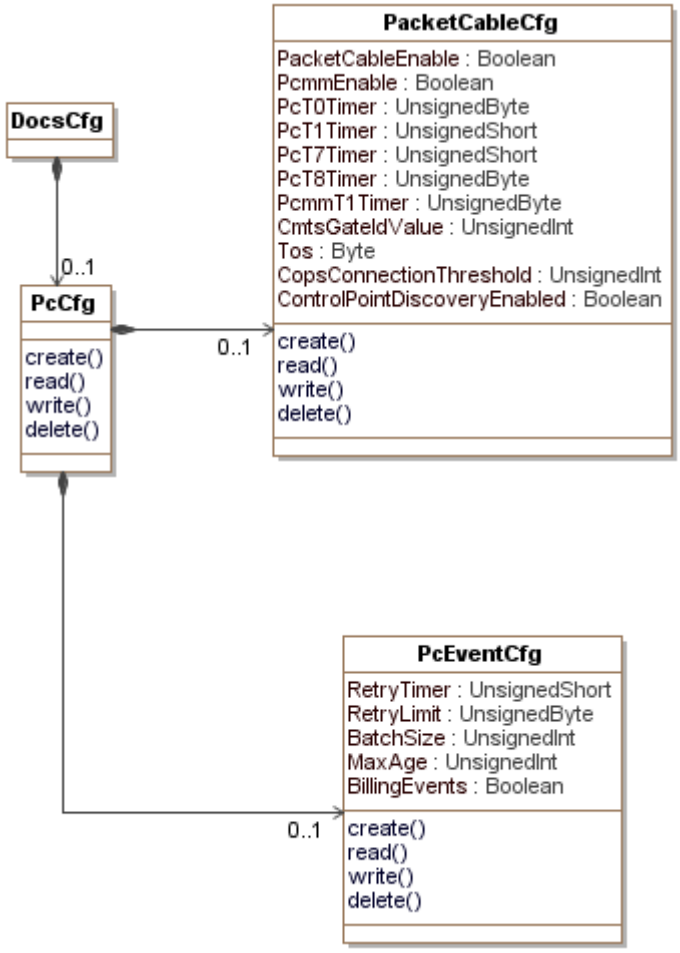


Figure 6.15: IPCablecom Configuration Objects

6.5.7.10.1 DocsCfg

This configuration object is included in figure 6.15 for reference. It is defined in clause 6.5.7.1.2.

6.5.7.10.2 PcCfg

The PcCfg object is the container for the IPCablecom and PCMM configuration objects. It has the following associations:

Table 6.155: PcCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
PacketCableConfig	Directed composition from PacketCableConfig		0..1	
PcEventCfg	Directed composition from PcEventCfg		0..1	

6.5.7.10.3 IPCablecomConfig Object

This object is used for configuring IPCablecom and PCMM services on the CCAP.

Table 6.156: PacketCableConfig Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
PacketCableEnable	Boolean	No			false
PcmmEnable	Boolean	No			false
PcT0timer	UnsignedByte	No		seconds	30
PcT1Timer	UnsignedShort	No		seconds	200
PcT7Timer	UnsignedShort	No		seconds	200
PcT8Timer	UnsignedByte	No		seconds	0
PcmmT1Timer	UnsignedByte	No		seconds	200
CmtsGateIdValue	UnsignedInt	Yes	0..16 383		
Tos	Byte	Yes	-1 | 0..63		
CopsConnectionThreshold	UnsignedInt	Yes		connections/15 mins	
ControlPointDiscoveryEnabled	Boolean	No			false

6.5.7.10.3.1 PacketCableConfig Object Attributes

6.5.7.10.3.1.1 PacketCableEnable

This configuration attribute allows the operator to enable IPCablecom services on the CCAP.

6.5.7.10.3.1.2 PcmmEnable

This configuration attribute allows the operator to enable IPCablecom Multimedia services on the CCAP.

6.5.7.10.3.1.3 PcT0Timer

This configuration attribute allows the operator to define the value in seconds for the IPCablecom T0 timer.

6.5.7.10.3.1.4 PcT1Timer

This configuration attribute allows the operator to define the value in seconds for the IPCablecom T1 timer.

6.5.7.10.3.1.5 PcT7Timer

This attribute allows for the setting of the Timeout for Admitted QoS Parameters for the service flow to the value specified for this timer. In the case of a flow with multiple sub-flows, the flow's Timeout for Admitted QoS Parameters is set to the value of timer T7 from the most recently received Gate-Set message for any subflow on the flow. The Timeout for Admitted QoS Parameters limits the period of time that the CMTS holds resources for a service flow's Admitted QoS Parameter Set while they are in excess of its Active QoS Parameter Set.

The recommended default value of this timer is 200 seconds.

6.5.7.10.3.1.6 PcT8Timer

This attribute configures the Timeout for Active QoS Parameters for the service flow to the value specified for this timer. In the case of a flow with multiple sub-flows, the flow's Timeout for Active QoS Parameters is set to the value of timer T8 from the most recently received Gate-Set message for any sub-flow on the flow. The Timeout for Active QoS Parameters limits the period of time resources remain unused on an active service flow.

6.5.7.10.3.1.7 PcmmT1Timer

This configuration attribute allows the operator to define the value in seconds for the IPCablecom Multimedia T1 timer.

6.5.7.10.3.1.8 CmtsGateIdValue

This configuration attribute allows the operator to define the value for the CMTS ID portion of PCMM GateIds. This value is the 13 least significant bits (0-12) of the GateId.

6.5.7.10.3.1.9 Tos

This configuration attribute allows the operator to define the value for the Tos bits in outgoing COPS messages.

6.5.7.10.3.1.10 CopsConnectionThreshold

This configuration attribute allows the operator to define the threshold number of COPS connections per 15-minute interval.

6.5.7.10.3.1.11 ControlPointDiscoveryEnabled

This attribute enables or disables the Control Point Discovery functionality described in the IPCablecom Specifications. The default value is false.

6.5.7.10.4 PcEventCfg Object

This object configures event messaging for IPCablecom.

Table 6.157: PcEventCfg Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
RetryTimer	UnsignedShort	No	10..10 000	milliseconds	3 000
RetryLimit	UnsignedByte	No	0..9		3
BatchSize	UnsignedInt	Yes			
MaxAge	UnsignedInt	Yes		seconds	
BillingEvents	Boolean	No			false

6.5.7.10.4.1 PcEventCfg Object Attributes

6.5.7.10.4.1.1 RetryTimer

This configuration attribute allows the configuration of the number of seconds the CCAP should wait before sending a message that was not acknowledged.

6.5.7.10.4.1.2 RetryLimit

This configuration attribute allows the configuration of the number of times the CCAP should retry before sending a message.

6.5.7.10.4.1.3 BatchSize

This configuration attribute allows the configuration of the number of records the CCAP should bundle in a single message to a billing or Record Keeping Server (RKS).

6.5.7.10.4.1.4 MaxAge

This object defines the max age of messages to be sent to an RKS or billing server.

6.5.7.10.4.1.5 BillingEvents

This attribute tells the CCAP if it needs to send billing events to a billing server/RKS.

6.5.7.11 Load Balance Configuration Objects

This section defines the configuration objects needed for configuring DOCSIS load balancing on the CCAP. These objects are based upon the Load Balancing Objects defined in Annex I of [7].

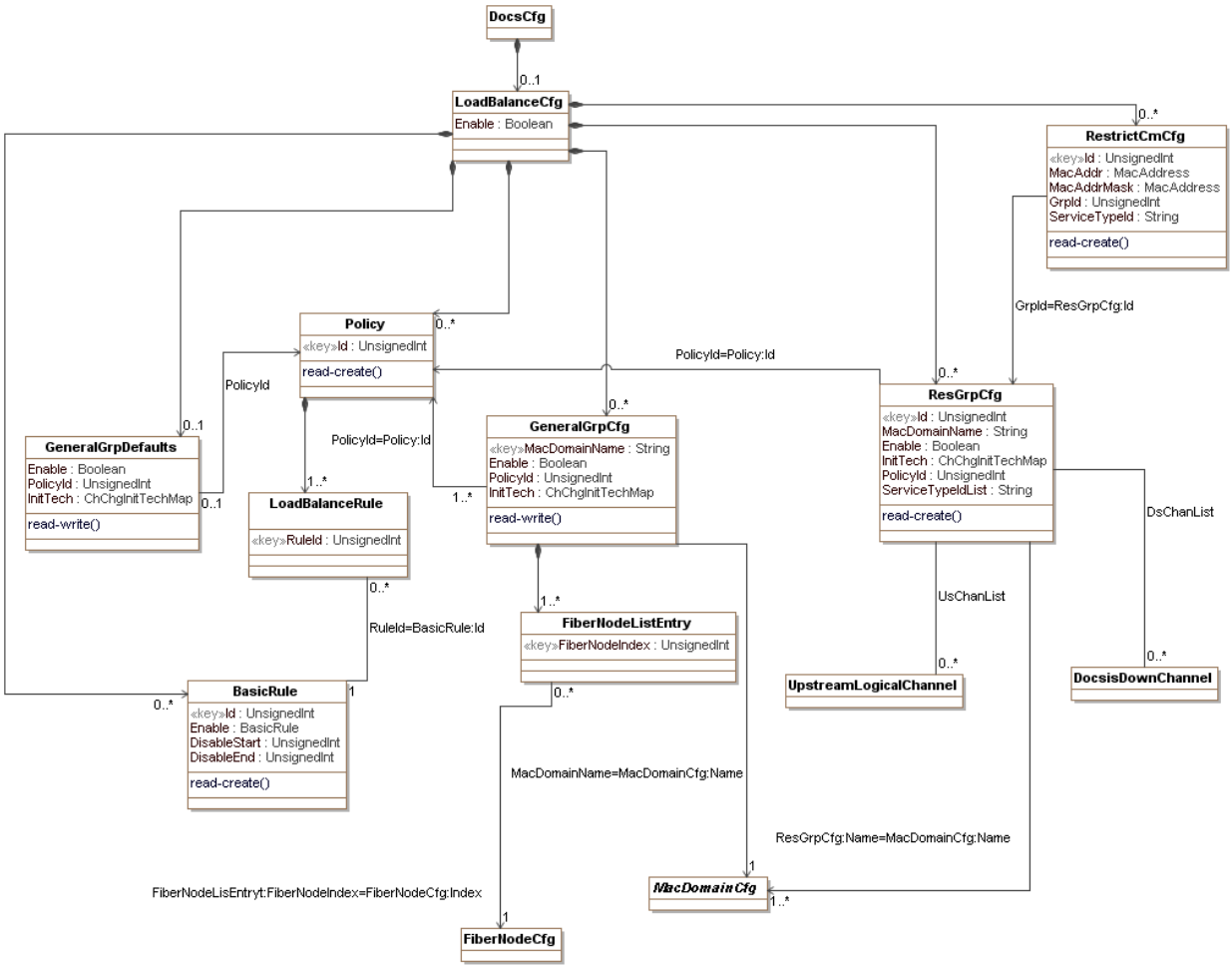


Figure 6.16: Load Balance Configuration Objects

6.5.7.11.1 DocsCfg

This configuration object is included in figure 6.15 for reference. It is defined in clause 6.5.7.1.2.

6.5.7.11.2 LoadBalanceCfg

This object enables and disables Autonomous Load Balancing Operations. It is based on the System object and is used with the following modification: The EnableError attribute has been removed because it does not provide enough information about what aspect of the configuration has caused enabling to fail.

Reference: [7], System Object.

Table 6.158: LoadBalanceCfg Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Enable	Boolean	No			true

Table 6.159: LoadBalanceCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
GeneralGrpCfg	Directed composition to GeneralGrpCfg		0..*	
GeneralGrpDefaults	Directed composition to GeneralGrpDefaults		0..1	
BasicRule	Directed composition to BasicRule		0..*	
Policy	Directed composition to Policy		0..*	
ResGrpCfg	Directed composition to ResGrpCfg		0..*	
RestrictCmCfg	Directed composition to RestrictCmCfg		0..*	

6.5.7.12.1 LoadBalanceCfg Object Attributes

6.5.7.12.1.1 Enable

This attribute when set to 'true' enables Autonomous Load Balancing operation on the CCAP; otherwise Autonomous Load Balancing is disabled.

When Autonomous Load Balancing is enabled, the CCAP may reject Externally-Directed Load Balancing operations. However, even when Autonomous Load Balancing is disabled, the CCAP is required to assign load balancing parameters to CMs as provisioned in the configuration file and/or RestrictCmCfg object.

6.5.7.11.3 GeneralGrpCfg

This object allows configuration of load balancing parameters for General Load Balancing Groups by way of MAC Domain-Fiber Node pairs. In many deployments, a MAC Domain-Fiber Node pair will equate to an MD-CM-SG (which always equates to a General Load Balancing Group [GLBG]). In the case where an MD-CM-SG spans multiple Fiber Nodes, there will be multiple instances of this object that represent the General Load Balancing Group (MD-CM-SG); the CCAP shall enforce that such instances all have the same attribute values. Any time a fiber node is associated to a MAC Domain, an instance of this object is defined by the CCAP and populated with either the same values as the other fiber nodes associated with the same MD-CM-SG (if any exist) or default values from the GeneralGrpDefaults object. Similarly, when a fiber node is no longer paired with a MAC Domain, the corresponding instance is deleted from the object.

Reference: [7], GeneralGrpCfg Object.

Table 6.160: GeneralGrpCfg Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
MacDomainName	String	Yes (key)			
Enable	Boolean	No			
PolicyId	UnsignedInt	No			
InitTech	EnumBits	No	reinitializeMac(0), broadcastInitRanging(1), unicastInitRanging(2), initRanging(3), direct(4)		

Table 6.161: GeneralGroupCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
Policy	Association to Policy	0..1		PolicyId=Policy:Id
FiberNodeCfg	Directed association to FiberNodeCfg	0..*	1	FiberNodeIndex= FiberNodeCfg:Index
MacDomainCfg	Directed association to MacDomainCfg		1	MacDomainName= MacDomainCfg:Name
FiberNodeListEntry	Directed composition to FiberNodeListEntry		1..*	

6.5.7.11.3.1 GeneralGrpDefaults Object Attributes

6.5.7.11.3.1.1 MacDomainName

This key configures the MAC Domain being associated with a list of fiber nodes.

6.5.7.11.3.1.2 Enable

This attribute, when set to 'true', enables Autonomous Load Balancing for the General Load Balancing Group associated with this instance. When set to 'false', Autonomous Load Balancing is disabled.

6.5.7.11.3.3 PolicyId

This attribute defines the default load balancing policy for the General Load Balancing Group associated with this instance. The value 0 is reserved to indicate no policy is associated with this GeneralGrpCfg instance.

6.5.7.11.3.1.4 InitTech

This attribute defines the load balancing initialization technique for the General Load Balancing Group associated with this instance.

Each bit position represents the internal associated technique as described below:

- reinitializeMac: Reinitialize the MAC.
- broadcastInitRanging: Perform Broadcast initial ranging on new channel before normal operation.
- unicastInitRanging: Perform unicast ranging on new channel before normal operation.
- initRanging: Perform either broadcast or unicast ranging on new channel before normal operation.
- direct: Use the new channel(s) directly without re-initializing or ranging.

Multiple bits can be set to 1 to allow the CCAP to select the most suitable technique in a proprietary manner.

A value with all bits '0' means no channel changes allowed.

References: [6], Initialization Technique.

6.5.7.11.4 FiberNodeListEntry

This object configures an entry in the list of fiber node names that are associated with the configured MAC Domain.

Table 6.162: FiberNodeListEntry Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
FiberNodeIndex	UnsignedInt	Yes (key)			

Table 6.163: FiberNodeListEntry Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
FiberNodeCfg	Directed composition to FiberNodeCfg	0..*	1	FiberNodeListEntry: FiberNodeIndex= FiberNodeCfg:Index

6.5.7.11.4.1 FiberNodeListEntry Object Attributes

6.5.7.11.4.1.1 FiberNodeIndex

This attribute configures the Index of a FiberNode instance associated with the load balancing group.

6.5.7.11.5 GeneralGrpDefaults

This object provides the default load balancing parameters for General Load Balancing Groups (MD-CM-SGs) that are used when instances of GeneralGrpCfg are created by the CCAP.

Reference: [7], GeneralGrpDefaults Object.

Table 6.164: GeneralGrpDefaults Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Enable	Boolean	No			true
PolicyId	UnsignedInt	No			0
InitTech	EnumBits	No	reinitializeMac(0), broadcastInitRanging(1), unicastInitRanging(2), initRanging(3), direct(4)		'F8'H

Table 6.165: GeneralGrpDefaults Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
Policy	Association to Policy	0..1		PolicyId

6.5.7.11.5.1 GeneralGrpDefaults Object Attributes

6.5.7.11.5.1.1 Enable

This attribute represents the default value for the Enable attribute of the GeneralGrpCfg object.

6.5.7.11.5.1.2 PolicyId

This attribute represents the default value for the PolicyId attribute of the GeneralGrpCfg object. The value 0 is reserved to indicate no policy is associated with the GeneralGrpDefaults object.

6.5.7.11.5.1.3 InitTech

This attribute represents the default value for the InitTech attribute of the GeneralGrpCfg object.

6.5.7.11.6 BasicRule

This object represents a basic rule set applicable to a load balancing policy that references it.

Reference: [7], BasicRule Object.

Table 6.166: BasicRule Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Id	UnsignedInt	Yes (key)			
Enable	Enum	No	other(1), enabled(2), disabled(3), disabledPeriod(4)		disabled
DisableStart	UnsignedInt	No	0..86 399		0
DisableEnd	UnsignedInt	No	0..86 399		0

Table 6.167: BasicRule Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
LoadBalanceRule	Association to LoadBalanceRule	1	0..*	RuleId=BasicRule:Id

6.5.7.11.6.1 BasicRule Object Attributes

6.5.7.11.6.1.1 Id

This key configures a unique identifier of a load balancing rule set for this object.

6.5.7.11.6.1.2 Enable

This attribute when set to 'enabled' enables Autonomous Load Balancing (independently of the load balancing group enable/disable state). The rule set is disabled if set to 'disabled'. If set to 'disabledPeriod', the rule set is disabled during a period of time configured in the DisableStart and DisableEnd attributes.

6.5.7.11.6.1.3 DisableStart

This attribute disables load balancing from the time stated by this attribute when the attribute Enable is set to 'disablePeriod'. The time is defined in seconds since midnight. This attribute is required if the value of the Enable attribute is disabledPeriod; otherwise it is ignored.

6.5.7.11.6.1.4 DisableEnd

This attribute disables load balancing until the time stated by this attribute when the attribute Enable is set to 'disablePeriod'. The time is defined in seconds since midnight. This attribute is required if the value of the Enable attribute is disabledPeriod; otherwise it is ignored.

6.5.7.11.7 Policy

This object describes the set of load balancing policies. All the rules contained in a load balancing policy apply to Autonomous Load Balancing operations. Load balancing rules are defined within the present document or can be vendor-defined as well.

Reference: [7], Policy Object.

Table 6.168: Policy Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Id	UnsignedInt	Yes (key)	1.. 4 294 967 295		

Table 6.169: Policy Object Associations

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

6.5.7.11.7.1 Policy Object Attributes

6.5.7.11.7.1.1 Id

This key configures a unique identifier for this load balancing policy.

6.5.7.11.8 LoadBalanceRule

This object allows a load balancing rule to be associated with a Policy instance.

Table 6.170: LoadBalanceRule Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
RuleId	UnsignedInt	Yes (key)			

6.5.7.11.8.1 LoadBalanceRule Object Attributes

6.5.7.11.81.1 RuleId

This key configures a unique identifier for this instance.

6.5.7.11.9 ResGrpCfg

This object represents the configuration of Restricted Load Balancing Groups.

Reference: [7], ResGrpCfg Object.

Table 6.171: ResGrpCfg Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Id	UnsignedInt	Yes (key)			
MacDomainName	String	Yes			
Enable	Boolean	No			true
InitTech	EnumBits	No	reinitializeMac(0), broadcastInitRanging(1), unicastInitRanging(2), initRanging(3), direct(4)		'F8'H
PolicyId	UnsignedInt	No			0
ServiceTypeIdList	String	No	0-255		""

Table 6.172: ResGrpCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
Policy	Directed association to Policy	0..1		PolicyId=Policy:Id
UpstreamLogicalChannel	Association to UpstreamLogicalChannel		0..*	UsChanList
DocsisDownChannel	Association to DocsisDownChannel		0..*	DsChanList
MacDomainCfg	Directed association to MacDomainCfg		1..*	ResGrpCfg:Name= MacDomainCfg:Name

6.5.7.11.9.1 ResGrpCfg Object Attributes

6.5.7.11.9.1.1 Id

This key configures a unique index assigned to the Restricted Load Balancing Group by the user for provisioning purposes. This value is unique within a CCAP and is matched with the CM signaled Load Balancing Group ID TLV value when determining the CM Load Balancing Group assignment based on such TLV value.

References: [6], Channel Assignment During Registration section.

6.5.7.11.9.1.2 MacDomainName

This attribute configures the MAC domain where the Restricted Load balancing Group applies. A zero length string indicates that vendor-specific mechanisms are used to define the Restricted Load Balancing Group. For example, to provide Load Balancing Groups across MAC domains.

6.5.7.11.9.1.3 Enable

This attribute when set to 'true' enables Autonomous Load Balancing on this Restricted Load Balancing Group. The value 'false' disables the load balancing operation on this group.

6.5.7.11.9.1.4 InitTech

This attribute represents the initialization techniques that the CCAP can use to load balance cable modems in the Load Balancing Group.

Each bit position represents the internal associated technique as described below:

- reinitializeMac: Reinitialize the MAC.
- broadcastInitRanging: Perform Broadcast initial ranging on new channel before normal operation.
- unicastInitRanging: Perform unicast ranging on new channel before normal operation.
- initRanging: Perform either broadcast or unicast ranging on new channel before normal operation.
- direct: Use the new channel(s) directly without re-initializing or ranging.

By default this object is initialized with all the defined bits having a value of '1'.

Multiple bits can be set to 1 to allow the CCAP to select the most suitable technique in a proprietary manner.

A value with all bits '0' means no channel changes allowed.

References: EN 302 878-4 [6], Initialization Technique.

6.5.7.11.9.1.5 PolicyId

This attribute represents the default load balancing policy of this Restricted Load Balancing Group. A policy is described by a set of conditions (rules) that govern the load balancing process for a cable modem. The CCAP assigns this Policy ID value to a cable modem associated with the group ID when the cable modem does not signal a Policy ID during registration. The Policy ID value is intended to be a numeric reference to an instance of the Policy object. The Policy ID of value 0 is reserved to indicate no policy is associated with the load balancing group.

6.5.7.11.9.1.6 ServiceTypeIdList

This attribute represent a space separated list of ServiceType IDs that will be compared against the cable modem provisioned Service Type ID to determine the most appropriate Restricted Load Balancing Group.

References: EN 302 878-4 [6], Channel Assignment During Registration section.

6.5.7.11.10 RestrictCmCfg

This object configures a list of cable modems being statically provisioned at the CCAP to a Restricted Load Balancing Group.

Reference: [7], RestrictCmCfg Object.

Table 6.173: RestrictCmCfg Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Id	UnsignedInt	Yes (key)			
MacAddr	MacAddress	No			'000000000000'H
MacAddrMask	MacAddress	No			
GrpId	UnsignedInt	No			
ServiceTypeId	String	No	0-16		""

Table 6.174: RestrictCmCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
ResGrpCfg	Directed association to ResGrpCfg			GrpId=ResGrpCfg:Id

6.5.7.11.10.1 RestrictCmCfg Object Attributes

6.5.7.11.10.1.1 Id

This key represents the unique identifier of an instance of this object. The CCAP maintains a unique instance per MAC Address/MAC Address Mask combination.

6.5.7.11.10.1.2 MacAddr

This attribute represents the MAC Address of the cable modem within the Restricted Load Balancing Group.

6.5.7.11.10.1.3 MacAddrMask

This attribute corresponds to a bit mask acting as a wild card to associate a cable modem MAC addresses to a Restricted Load Balancing Group ID referenced by a restricted group Id or a Service Type ID. The cable modem matching criteria is performed by bit-ANDed the cable modem MAC address with the MacAddrMask attribute and being compared with the bit-ANDed of attributes MacAddr and MacAddrMask. A cable modem MAC address look up is performed first with instances containing this attribute value not null; if several entries match, the largest consecutive bit match from MSB to LSB is used. Empty value is equivalent to the bit mask all in ones.

6.5.7.11.10.1.4 GrpId

This attribute represents the Restricted Load Balancing Group identifier of this entry associated with the cable modem MAC address - MAC address mask combination. If this attribute is not configured, this instance is matched only against the ServiceTypeId value.

6.5.7.11.10.1.5 ServiceTypeId

This attribute represents the Service Type Id associated with this cable modem MAC address - MAC Address mask combination. If this attribute is not configured, this instance is matched only against the GrpId value; if both GrpId and this attribute are not present, the instance is ignored for matching purposes.

6.5.8 CCAP Network Configuration Objects

This section is a collection of configuration objects that are specific to the chassis and not to DOCSIS or video services on a CCAP.

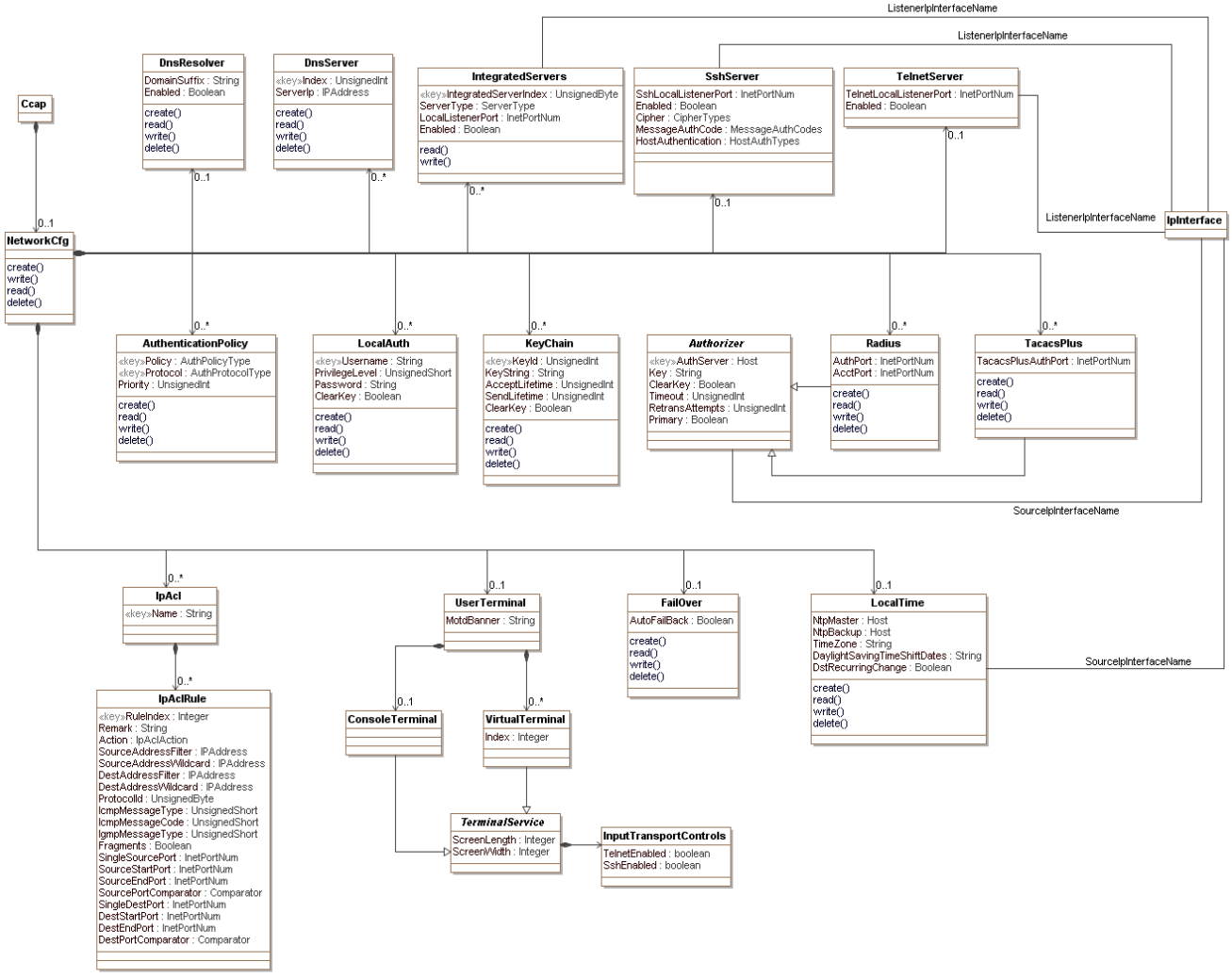


Figure 6.17: CCAP Network Configuration Objects

6.5.8.1 Ccap

This configuration object is included in figure 6.17 for reference. It is defined in clause 6.5.4.1.

6.5.8.2 NetworkCfg

The NetworkCfg object is the primary container of network configuration objects. It has the following associations:

Table 6.175: NetworkCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
DnsResolver	Directed composition to DnsResolver		0..1	
DnsServer	Directed composition to DnsServer		0..*	
IntegratedServers	Directed composition to Integrated Servers		0..*	
SshServer	Directed composition to SshServer		0..1	
TelnetServer	Directed composition to TelnetServer		0..1	
AuthenticationPolicy	Directed composition to AuthenticationPolicy		0..*	
LocalAuth	Directed composition to LocalAuth		0..*	
Radius	Directed composition to Radius		0..*	
TacacsPlus	Directed composition to TacacsPlus		0..*	
KeyChain	Directed composition to KeyChain		0..*	
IpAcl	Directed composition to IpAcl		0..*	
UserTerminal	Directed composition to UserTerminal		0..1	
FailOver	Directed composition to FailOver		0..1	
LocalTime	Directed composition to Time		0..1	

6.5.8.3 DnsResolver

This object allows the configuration of DNS servers and the configuration of default domain suffix information. The objects in this configuration object are scalars.

Table 6.176: DnsResolver Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
DomainSuffix	String	Yes			
Enabled	Boolean	No			true

6.5.8.3.1 DnsResolver Object Attributes

6.5.8.3.1.1 DomainSuffix

The attribute DomainSuffix configures a Domain Suffix that should be post-pended to any hostname lookup that does not consist of a Fully Qualified Domain Name (FQDN).

6.5.8.3.1.2 Enabled

This attribute configures if the associated domain suffix should be applied to hostnames that do not include an FQDN.

6.5.8.4 DnsServer

This object allows the configuration of the different DNS Servers that the CCAP can use to get Domain Name Resolution.

Table 6.177: DnsServer Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes (Key)			
ServerIp	IpAddress	Yes			

6.5.8.4.1 DnsServer Object Attributes

6.5.8.4.1.1 Index

This attribute configures the index for this instance of DnsServer.

6.5.8.4.1 ServerIp

This attribute configures the IP address of the DNS server used by the CCAP for DNS resolution. No distinction is made for IPv6 or IPv4 addresses here.

6.5.8.5 IntegratedServers

This configuration object defines the types of servers integrated into the CCAP and their respective administrative states. At run time an object for each server type will be instantiated with its IANA-defined default port; see [8]. To define a different default port, the operator will update the existing IntegratedServers object for that server type with the new port number specified.

Table 6.178: IntegratedServers Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
IntegratedServerIndex	UnsignedByte	Yes (Key)			
ServerType	Enum	Yes	other(1), ftp(2), http(3), netconf(4)		
LocalListenerPort	InetPortNum	No			See attribute description
Enabled	Boolean	No			false

Table 6.179: IntegratedServers Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
IpInterface	Association with an IpInterface			ListenerIpInterfaceName

When an IP interface is selected, this specifies the IP interface on which the server listens. If an IP interface is not specified, the behavior of the CCAP is vendor specific.

6.5.8.5.1 IntegratedServers Object Attributes

6.5.8.5.1.1 IntegratedServerIndex

This attributes configures a unique identifier for this IntegratedServers instance.

6.5.8.5.1.2 ServerType

This attribute configures the type of server being configured on the CCAP. The value of other(1) is used when a vendor-extension has been implemented for this attribute. The CCAP may support a NETCONF server-type option.

6.5.8.5.1.3 LocalListenerPort

This attribute configures the TCP or UDP port number on which the server listens. The CCAP shall assign the default value as the IANA-assigned port number associated with the ServerType selected, as defined in [8].

6.5.8.5.1.4 Enabled

This attribute configures the running state of the server. True means that the server will actively listen on the specified port. False means that the specific server is disabled.

6.5.8.6 SshServer

This configuration object defines an integrated SSHv2 server in the CCAP. The CCAP SSH server shall support SSH version 2 as defined in:

- RFC 4250 [15]
- RFC 4251 [16]
- RFC 4252 [17]
- RFC 4253 [18]
- RFC 4254 [19]

This configuration object allows different combinations of cipher, message authentication code, and host authentication code to be configured; however, a CCAP might not support all possible combinations of these three attributes.

Table 6.180: SshServer Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
SshLocalListenerPort	InetPortNum	No			22
Enabled	Boolean	No			false
Cipher	EnumBits	No	other(0), 3des(1), aes128(2), aes192(3), aes256(4), arcfour(5), blowfish(6), cast(7), twofish128(8), twofish192(9), twofish256(10)	[16] [17]	3des
MessageAuthCode	EnumBits	No	other(0), md5(1), md5-96(2), sha1(3), sha1-96(4), ripemd-160(5), sha2-256(6), sha2-512(7)	[19]	vendor-specific
HostAuthentication	Enum	No	other(0), none(1), ssh-dss(2), ssh-rsa(3), pgp-sign-rsa(4), pgp-sign-dss(5)		None

Table 6.181: SshServer Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
IpInterface	Association with an IpInterface			ListenerIpInterfaceName

When an IP interface is selected, this specifies the IP interface on which the server listens. If an IP interface is not specified, the behavior of the CCAP is vendor specific.

6.5.8.6.1 SshServer Object Attributes

6.5.8.6.1.1 LocalListenerPort

This object configures the TCP or UDP port number on which the server listens.

6.5.8.6.1.2 Enabled

This attribute configures the running state of the server. True means that the server will actively listen on the specified port. False means that the specific server is disabled.

6.5.8.6.1.3 Cipher

This attribute configures the set of encryption algorithms that are allowed on the SSH interface. SSH will use the enabled set of algorithms to negotiate the algorithm to use with the connecting client. The CCAP system shall log an error if the configuration file enables a cipher algorithm that is not supported. The bit setting of "other" can be used to enable an algorithm supported by the CCAP that is not in the defined list.

6.5.8.6.1.4 MessageAuthCode

This attribute configures the set of message authentication algorithms that are allowed on the SSH interface. SSH will use the enabled set of algorithms to negotiate the algorithm to use with the connecting client to ensure message integrity. The CCAP system shall log an error if the configuration file enables a MAC algorithm that is not supported. The bit setting of "other" can be used to enable an algorithm supported by the CCAP that is not in the defined list.

6.5.8.6.1.5 HostAuthentication

This attribute enables SSH host authentication using public keys in a specified format. It is assumed that user authentication will be configured in the same way as other CCAP interfaces. The file format for key storage is outside the scope of the present document.

6.5.8.7 TelnetServer

This configuration object defines an integrated Telnet server in the CCAP.

Table 6.182: TelnetServer Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
TelnetLocalListenerPort	InetPortNum	No			23
Enabled	Boolean	No			false

Table 6.183: TelnetServer Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
IpInterface	Association with an IpInterface			ListenerIpInterfaceName

When an IP interface is selected, this specifies the IP interface on which the server listens. If an IP interface is not specified, the behavior of the CCAP is vendor specific.

6.5.8.7.1 TelnetServer Object Attributes

6.5.8.7.1.1 LocalListenerPort

This object configures the TCP or UDP port number on which the server listens.

6.5.8.7.1.2 Enabled

This attribute configures the running state of the server. True means that the server will actively listen on the specified port. False means that the specific server is disabled.

6.5.8.8 AuthenticationPolicy

This configuration object allows the configuration of authentication policy. The Priority attribute controls which service is used first for authenticating users.

Table 6.184: AuthenticationPolicy Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Policy	Enum	Yes (Key)	other(1), login(2), privilegedMode(3)		
Protocol	Enum	Yes (Key)	other(1), radius(2), tacacsPlus(3), localAuth(4), none(5)		
Priority	UnsignedInt	Yes			

6.5.8.8.1 AuthenticationPolicy Object Attributes

6.5.8.8.1.1 Policy

This attribute is the first part of the key and configures the policy type for the specified protocol. The privilegedMode(3) option is an administrative role that allows the user to execute all available commands. The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.8.8.1.2 Protocol

This attribute is the second part of the key and represents the protocol used for authentication. The value of other(1) is used when a vendor extension has been implemented for this attribute.

6.5.8.8.1.3 Priority

This attribute sets a priority for the protocol selected. Higher numbers are higher priority. A specified policy cannot have the same priority across multiple protocols.

6.5.8.9 LocalAuth

This object configures the local user accounts and privilege levels.

Table 6.185: LocalAuth Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Username	String	Yes (Key)			
PrivilegeLevel	UnsignedShort	Yes	0..15		
Password	String	Yes			
ClearKey	Boolean	Yes			

6.5.8.9.1 LocalAuth Object Attributes

6.5.8.9.1.1 UserName

This attribute configures the "login" name to be used.

6.5.8.9.1.2 PrivilegeLevel

This attribute correspond to the user's privilege level. The highest number provides the most user privileges.

6.5.8.9.1.3 Password

This attribute correspond to the user's password. Upon export, the CCAP shall export the Password attribute of the LocalAuth object encrypted with a vendor-specific algorithm.

6.5.8.9.1.4 ClearKey

This attribute indicates whether the Password attribute is included in the XML configuration file in the clear (true) or encrypted (false). This attribute defines the status of the password (encrypted or decrypted), not whether the device should export the password in the clear or encrypted. Regardless of the value of this setting, the password will always be exported as encrypted.

6.5.8.10 Authorizer

The Authorizer abstract class holds common attributes used for configuring TACACS+ and Radius services for the CCAP.

Table 6.186: Authorizer Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
AuthServer	Host	Yes (Key)			
Key	String	Yes			
ClearKey	Boolean	Yes			
Timeout	Byte	No		seconds	3
RetransAttempts	UnsignedByte	No			1
Primary	Boolean	No			false

Table 6.187: Authorizer Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
IpInterface	Association with an IpInterface			SourceIpInterfaceName

This association specifies the IP interface to use as the source interface. If an IP interface is not specified, the behavior of the CCAP is vendor specific.

6.5.8.10.1 Authorizer Object Attributes

6.5.8.10.1.1 AuthServer

This attribute is the IPv4 address or FQDN of the server.

6.5.8.10.1.2 Key

This attribute corresponds to the shared secret that is used to encrypt the communication.

Upon export, the CCAP shall export the Key attribute of the TacacsPlus object encrypted with a vendor-specific algorithm.

6.5.8.10.1.3 ClearKey

This attribute indicates whether the Key attribute is included in the XML configuration file in the clear (true) or encrypted (false). This attribute defines the status of the key (encrypted or decrypted), not whether the device should export the key in the clear or encrypted. Regardless of the value of this setting, the key will always be exported as encrypted.

6.5.8.10.1.4 Timeout

This attribute defines the number of seconds before a connection is declared inactive.

6.5.8.10.1.5 RetransAttempts

This attribute defines the number of retransmissions before giving up the connection.

6.5.8.10.1.6 Primary

This attribute designates whether this TACACS instance is the primary or backup server.

6.5.8.11 Radius

This configuration object creates the configuration for Radius servers.

Table 6.188: Radius Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
AuthPort	InetPortNum	No	0..65 535		1 812
AcctPort	InetPortNum	No	0..65 535		1 813

Table 6.189: Radius Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
Authorizer	Specialization of Authorizer			

6.5.8.11.1 Radius Object Attributes

6.5.8.11.1.1 AuthPort

This attribute defines the TCP port on which AAA authentication and authorization are performed.

6.5.8.11.1.2 AcctPort

This attribute defines the TCP port on which AAA accounting is performed.

6.5.8.12 TacacsPlus

This configuration object configures TACACS+ services for the CCAP.

Table 6.190: TacacsPlus Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
TacacsPlusAuthPort	InetPortNum	No	0..65 535		49

Table 6.191: TacacsPlus Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
Authorizer	Specialization of Authorizer			

Specifies the IP interface to use as the source interface. If an IP interface is not specified, the behavior of the CCAP is vendor-specific.

6.5.8.12.1 TacacsPlus Object Attributes

6.5.8.12.1.1 TacacsPlusAuthPort

This attribute defines the TCP port used for communicating with the AAA server.

6.5.8.13 KeyChain

The KeyChain object allows the CCAP to be configured with different Ripv2 key change information.

Table 6.192: KeyChain Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
KeyId	UnsignedInt	Yes (Key)	0..2 147 483 647		
KeyString	String	Yes	1..79		
AcceptLifetime	UnsignedInt	Yes	0..2 147 483 647	seconds	
SendLifetime	UnsignedInt	No	0..2 147 483 647	seconds	0
ClearKey	Boolean	Yes			

6.5.8.13.1 KeyChain Object Attributes

6.5.8.13.1.1 KeyId

This attribute configures a KeyId used in RipV2 route updates.

6.5.8.13.1.2 KeyString

This attribute configures the actual key used for this instance. This value has to be the same on both the sender and receiver of the RIPv2 route.

6.5.8.13.1.3 AcceptLifetime

This attribute configures the accept lifetime value in seconds for the key in this instance.

6.5.8.13.1.4 SendLifetime

This attribute configures the send lifetime value in seconds for the key in this instance. A value of 0 (zero) means that there is no lifetime limit.

6.5.8.13.1.5 ClearKey

This attribute indicates whether the KeyString attribute is included in the XML configuration file in the clear (true) or encrypted (false). This attribute defines the status of the key (encrypted or decrypted), not whether the device should export the key in the clear or encrypted. Regardless of the value of this setting, the key will always be exported as encrypted.

6.5.8.14 IpAcl

This configuration object defines the attributes for the IP Access Control List object. This object defines an extended access control list.

Table 6.193: IpAcl Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Name	String	Yes (Key)	1..32		

Table 6.194: IpAcl Object Associations

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

6.5.8.14.1 IpAcl Object Attributes

6.5.8.14.1.1 Name

This attribute configures a unique identifier for an instance of this object.

6.5.8.15 IpAclRule

This configuration object defines an access control list rule contained within an IpAcl instance. Multiple rules can be contained within an IpAcl instance.

When the ACL rule is processed, the system will only match on the values configured in the rule. If an attribute is not provided in the configuration instance file, the CCAP will match any value for that attribute. For example, if ProtocolId is not specified, then any value for protocol Id in the packet will match the filter. If the CCAP rejects the configuration of an IpAclRule, the CCAP should also reject the IpAcl instance that contains the rule.

A configured instance of the IpAclRule object either holds a Remark or an Action. If it contains a Remark, then only the RuleIndex and Remark attributes are allowed. If the instance contains and Action, the Remark attribute is not allowed, but all other attributes can be included, as described in the following clauses.

Table 6.195: IpAclRule Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
RuleIndex	Integer	Yes (Key)			
Remark	String	No			
Action	Enum	No	other(1), deny(2), permit(3)		
SourceAddressFilter	IpAddress	No (note 1)			
SourceAddressWildcard	IpAddress	No (note 2)			
DestAddressFilter	IpAddress	No (note 1)			
DestAddressWildcard	IpAddress	No (note 3)			
ProtocolId	UnsignedByte	No			
IcmpMessageType	UnsignedShort	No	0..255		
IcmpMessageCode	UnsignedShort	No	0..255		
IgmpMessageType	UnsignedShort	No	0..255		
Fragments	Boolean	No			false
SingleSourcePort	InetPortNum	No			
SourceStartPort	InetPortNum	No			
SourceEndPort	InetPortNum	No			

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
SourcePortComparator	Enum	No	other(1), lessThan(2), greaterThan(3), equals(4), notEqual(5)		equals(4)
SingleDestPort	InetPortNum	No			
DestStartPort	InetPortNum	No			
DestEndPort	InetPortNum	No			
DestPortComparator	Enum	No	other(1), lessThan(2), greaterThan(3), equals(4), notEqual(5)		equals(4)
NOTE 1: If an Action is being configured, either SourceAddressFilter or DestAddressFilter is required for the configuration of this object, however both are not required. If an Action is configured and neither the SourceAddressFilter nor the DestAddressFilter value is provided in the configuration instance file, the CCAP shall reject the configuration of the IpAclRule instance. NOTE 2 If a SourceAddressFilter is configured, then the corresponding SourceAddressWildcard attribute also has to be configured. NOTE 3: If a DestAddressFilter is configured, then the corresponding DestAddressWildcard attribute also has to be configured.					

6.5.8.15.1 IpAclRule Object Attributes

6.5.8.15.1.1 RuleIndex

This attribute configures a unique identifier for the ACL rule. This value also sets the order in which rules are executed, with lower numbers executing first. The CCAP may restrict a range of indexes to a specific set of ACL attributes in a vendor-proprietary way.

6.5.8.15.1.2 Remark

This attribute provides a textual string that explains the intent of a group of ACL rules. When the Remark attribute is configured, only the RuleIndex attribute is allowed to be configured within that instance; if additional attributes are configured, the CCAP shall reject the configuration of the IpAclRule instance.

6.5.8.15.1.3 Action

This attribute configures the action the CCAP takes when the ACL rule matches a packet. This and all of the following attributes are only valid if a Remark attribute has not been configured.

6.5.8.15.1.4 SourceAddressFilter

This attribute defines an IP addresses to match the source address in the packet; it is used in conjunction with the SourceAddressWildcard attribute. The value can be an IPv4 or IPv6 address.

When both source and destination address filters are specified, each configured value has to be of the same IP type (either IPv4 or IPv6). If a DestAddressFilter is also specified, the CCAP shall reject the IpAclRule configuration if the address types do not match.

6.5.8.15.1.5 SourceAddressWildcard

The SourceAddressWildcard attribute defines which bits of the packet's source IP address are matched to the SourceAddressFilter attribute. The usage of the IP address wildcard differs from most typical applications where IP addresses are masked. Rather than restricting the defined IP address to a range of addresses by masking off the lowest significant bits of the address, the IP address mask is used as a wildcard.

Each bit in the SourceAddressWildcard set to zero indicates that the corresponding bit position in the packet's source IP address needs to exactly match the bit value in the corresponding bit position in the SourceAddressFilter. Each wildcard bit set to one indicates that both a zero bit and a one bit in the corresponding position of the packet's IP address will be considered a match to this access list entry. In other words, "ones" are places in bit positions that should be ignored. The set of "ones" does not have to start at LSB, nor has to cover consecutive bit positions. For example, a value of 0.0.255.1 is valid for an IPv4 wildcard.

For example, to configure the AclRule to match any IPv4 source address, a value of 0.0.0.0 would be configured in the SourceAddressFilter attribute and a value of 255.255.255.255 would be configured in the SourceAddressWildcard attribute.

A value of 0.0.0.0 for SourceAddressWildcard attribute signifies that the IP ACL will match packet to a specific host IP address specified in SourceAddressFilter attribute.

6.5.8.15.1.6 DestAddressFilter

This attribute defines an IP addresses to match the destination address in the packet; it is used in conjunction with the DestAddressWildcard attribute. The value can be an IPv4 or IPv6 address.

When both source and destination address filters are specified, each configured value has to be of the same IP type (either IPv4 or IPv6). If a SourceAddressFilter is also specified, the CCAP shall reject the IpAclRule configuration if the IP address types do not match.

6.5.8.15.1.7 DestAddressWildcard

The DestAddressWildcard attribute defines which bits of the packet's source IP address are matched to the DestAddressFilter attribute. The usage of the IP address wildcard differs from most typical applications where IP addresses are masked. Rather than restricting the defined IP address to a range of addresses by masking off the lowest significant bits of the address, the IP address mask is used as a wildcard.

The rules for matching are identical to those described for SourceAddressWildcard.

6.5.8.15.1.8 ProtocolId

This attribute defines an IP protocol number for the filter to match when the protocol is not ICMP or IGMP.

If the protocol is ICMP or IGMP, one of the following attributes will be configured instead:

- IcmpMessageType
- IgmpMessageType

6.5.8.15.1.9 IcmpMessageType

This attribute defines the ICMP message type for the filter to match. For the ICMP protocol, the ProtocolId attribute is not used. If both the ProtocolId and IcmpMessageType attributes are provided in an IpAclRule instance, the CCAP shall reject the configuration of the IpAclRule instance.

6.5.8.15.1.10 IcmpMessageCode

This attribute is only applicable if an IcmpMessageType has been configured. When this attribute is defined, the CCAP will filter packets that match the configured ICMP message type and message code. If the IcmpMessageCode attribute is provided in an IpAclRule instance, but the IgmpMessageType attribute is not, the CCAP shall reject the configuration of the IpAclRule instance.

6.5.8.15.1.11 IgmpMessageType

This attribute defines the IGMP message type for the filter to match. For the IGMP protocol, the ProtocolId attribute is not used. If both the ProtocolId and IgmpMessageType attributes are provided in an IpAclRule instance, the CCAP shall reject the configuration of the IpAclRule instance. If both the IcmpMessageType and IgmpMessageType attributes are provided in an IpAclRule instance, the CCAP shall reject the configuration of the IpAclRule instance.

6.5.8.15.1.12 Fragments

This attribute determines whether the ACL rule is applied to all fragments of a fragmented packet, or only to the initial fragment. A setting of false means that only the initial fragment is filtered.

6.5.8.15.1.13 SingleSourcePort

This attribute defines a single source port number for the ACL rule. The CCAP will filter a packet that comes from this source port.

For source port filtering, either the SingleSourcePort attribute, or the SourceStartPort and SourceEndPort attributes (i.e. a port range) is configured. If the SingleSourcePort and SourceStartPort attributes are provided in an IpAclRule instance, the CCAP shall reject the configuration of the IpAclRule instance.

6.5.8.15.1.14 SourceStartPort

This attribute defines the starting source port number for a range of ports defined for the ACL rule. When the SourceStartPort attribute is configured, the SourceEndPort attribute is also required. If the SourceStartPort attribute is provided in an IpAclRule instance, but a SourceEndPort attribute is not, the CCAP shall reject the configuration of the IpAclRule instance.

6.5.8.15.1.15 SourceEndPort

This attribute defines the ending source port number for a range of ports defined for the ACL rule. The value of this attribute has to be greater than the value in the SourceStartPort. If the SourceEndPort attribute is provided in an IpAclRule instance, but the SourceStartPort is not, the CCAP shall reject the configuration of the IpAclRule instance.

6.5.8.15.1.16 SourcePortComparator

This attribute defines how the filter matches a specified SingleSourcePort. This attribute is not valid if a SourceStartPort and SourceEndPort are provided. The filter can match if the source port number of the packet is less than, greater than, equal to, or not equal to the defined source port number.

The CCAP shall support the "less than", "greater than", and "not equal to" settings when a SingleSourcePort attribute is provided.

6.5.8.15.1.17 SingleDestPort

This attribute defines a single destination port number for the ACL rule. The CCAP will filter a packet that has this destination port.

For destination port filtering, either the SingleDestPort attribute, or the DestStartPort and DestEndPort attributes (i.e. a port range) are configured. If the SingleDestPort and DestStartPort attributes are provided in an IpAclRule instance, the CCAP shall reject the configuration of the IpAclRule instance.

6.5.8.15.1.18 DestStartPort

This attribute defines the starting destination port number for a range of ports defined for the ACL rule. When the DestStartPort attribute is configured, the DestEndPort attribute is also required. If the DestStartPort attribute is provided in an IpAclRule instance, but a DestEndPort attribute is not, the CCAP shall reject the configuration of the IpAclRule instance.

6.5.8.15.1.19 DestEndPort

This attribute defines the ending destination port number for a range of ports defined for the ACL rule. The value of this attribute has to be greater than the value in the DestStartPort. If the DestEndPort attribute is provided in an IpAclRule instance, but the DestStartPort is not, the CCAP shall reject the configuration of the IpAclRule instance.

6.5.8.15.1.20 DestPortComparator

This attribute defines how the filter matches a specified SingleDestPort. The filter can match if the destination port number of the packet is less than, greater than, equal to, or not equal to the defined destination port.

The CCAP shall support the "less than", "greater than", and "not equal to" settings when a SingleDestPort attribute is provided.

6.5.8.16 UserTerminal

This container object configures the user terminal instances for the CCAP, both the ConsoleTerminal instance and VirtualTerminal instances.

Table 6.196: UserTerminal Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
MotdBanner	String	No			""

Table 6.197: UserTerminal Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
TerminalService	Specialization of TerminalService			

6.5.8.16.1 UserTerminal Object Attributes

6.5.8.16.1.1 MotdBanner

This attribute configures the contents of a message of the day banner that displays to the user when the user logs into a virtual terminal.

6.5.8.17 VirtualTerminal

This object configures a virtual terminal interface on the CCAP.

Table 6.198: VirtualTerminal Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	Integer	Yes			

Table 6.199: VirtualTerminal Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
TerminalService	Specialization of TerminalService			

6.5.8.17.1 VirtualTerminal Object Attributes

6.5.8.17.1.1 Index

This attribute configures a unique index for this virtual terminal instance.

6.5.8.18 ConsoleTerminal

This object configures the console terminal interface on the CCAP.

Table 6.200: ConsoleTerminal Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
TerminalService	Specialization of TerminalService			

6.5.8.19 TerminalService

This abstract object holds attributes used to configure the console terminal and virtual terminal instances.

Table 6.201: TerminalService Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
ScreenLength	Integer	No		Lines	24
ScreenWidth	Integer	No		Columns	80

Table 6.202: TerminalService Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
InputTransportControls	Directed composition to InputTransportControls			

6.5.8.19.1 TerminalService Object Attributes

6.5.8.19.1.1 ScreenLength

This attribute configures the number of lines on the screen of the terminal instance.

6.5.8.19.1.2 ScreenWidth

This attribute configures the number of columns on the screen of the terminal instance.

6.5.8.20 InputTransportControls

This object configures SSH and Telnet settings for a virtual terminal instance.

Table 6.203: InputTransportControls Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
TelnetEnabled	Boolean	No			false
SshEnabled	Boolean	No			false

6.5.8.20.1 InputTransportControls Object Attributes

6.5.8.20.1.1 TelnetEnabled

This attribute configures whether Telnet is enabled on the virtual terminal interface.

6.5.8.20.1.2 SshEnabled

This attribute configures whether SSH is enabled on the virtual terminal interface.

6.5.8.21 FailOver

This object configures the automatic fail-over operation of the CCAP.

Table 6.204: FailOver Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
AutoFailBack	Boolean	No			true

6.5.8.21.1 FailOver Object Attributes

6.5.8.21.1.1 AutoFailBack

This attribute configures whether or not the CCAP automatically switches back to a line card after a failover event. If true, when the failed card is operational, the CCAP will begin using that card again. If False, the operator will have to perform the failback operation.

6.5.8.22 LocalTime

The LocalTime object allows the configuration of a Primary and Secondary NTP server, as well as other local time attributes. This object does not fully configure all NTP client parameters. Vendors may provide additional configuration objects to fully configure the NTP and SNTP protocols if desired.

Table 6.205: LocalTime Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
NtpMaster	Host	Yes			
NtpBackup	Host	No			
TimeZone	String	No			00
DaylightSavingTimeShiftDates	String	No			3.2.0/02.00, 11.1.0/02.00, 01
DstRecurringChange	Boolean	No			false

Table 6.206: LocalTime Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
IpInterface	Association with an IpInterface			SourceIpInterfaceName

This association specifies the IP interface to use as the source interface. If an IP interface is not specified, the behavior of the CCAP is vendor specific.

6.5.8.22.1 LocalTime Object Attributes

6.5.8.22.1.1 NtpMaster

This attribute configures the IP address or FQDN of the Master NTP server.

6.5.8.22.1.2 NtpBackup

This attribute configures the IP address or FQDN of the backup NTP Server in case the master NTP fails.

6.5.8.22.1.3 TimeZone

This attribute represents the offset value to the local time to arrive at UTC Time. The value has the following format:

hh[:mm] - the hour

(0 <= hh <= 24) - required, minutes

(0 <= mm <= 59) -the mm (minutes) is optional. The hour can be preceded by a minus sign (-).

6.5.8.22.1.4 DaylightSavingTimeShiftDates

This attribute indicates when to change to and from daylight saving (or summer) time. The value has the form: date1/time1,date2/time2,offset. The first date describes when the change from standard to daylight saving time occurs, and the second date describes when the change back happens.

Each time field describes when, in current local time, the change to the other time is made. The format of date is the following: m.w.d - The dth day (0 <= d <= 6) of week w of month m of the year (1 <= w <= 5, 1 <= m <= 12, where week 5 means "the last d day in month m", which may occur in the fourth or fifth week). Week 1 is the first week in which the dth day occurs. Day zero is Sunday.

The time format is the following: hh:mm - The offset value is the value that is added to the local time to arrive at UTC Time during the daylight saving time. The offset value has the following format: hh[:mm].

The default value is the second Sunday in March (start) and the first Sunday in November (end).

6.5.8.22.1.5 DstRecurringChange

This attribute controls whether the CCAP automatically adjusts the time to Daylight Saving Time (DST). If enabled, the CCAP will adjust the time based on the value of the DaylightSavingTimeCalendar attribute.

6.5.8.23 IpInterface

This configuration object is included in figure 6.17 for reference. It is defined in clause 6.5.9.5.

6.5.9 Interface Configuration Objects

Interfaces in the CCAP are different than ports, in that they are intended to be Layer 3 entities. The following object model shows the relationships for interfaces in the CCAP.

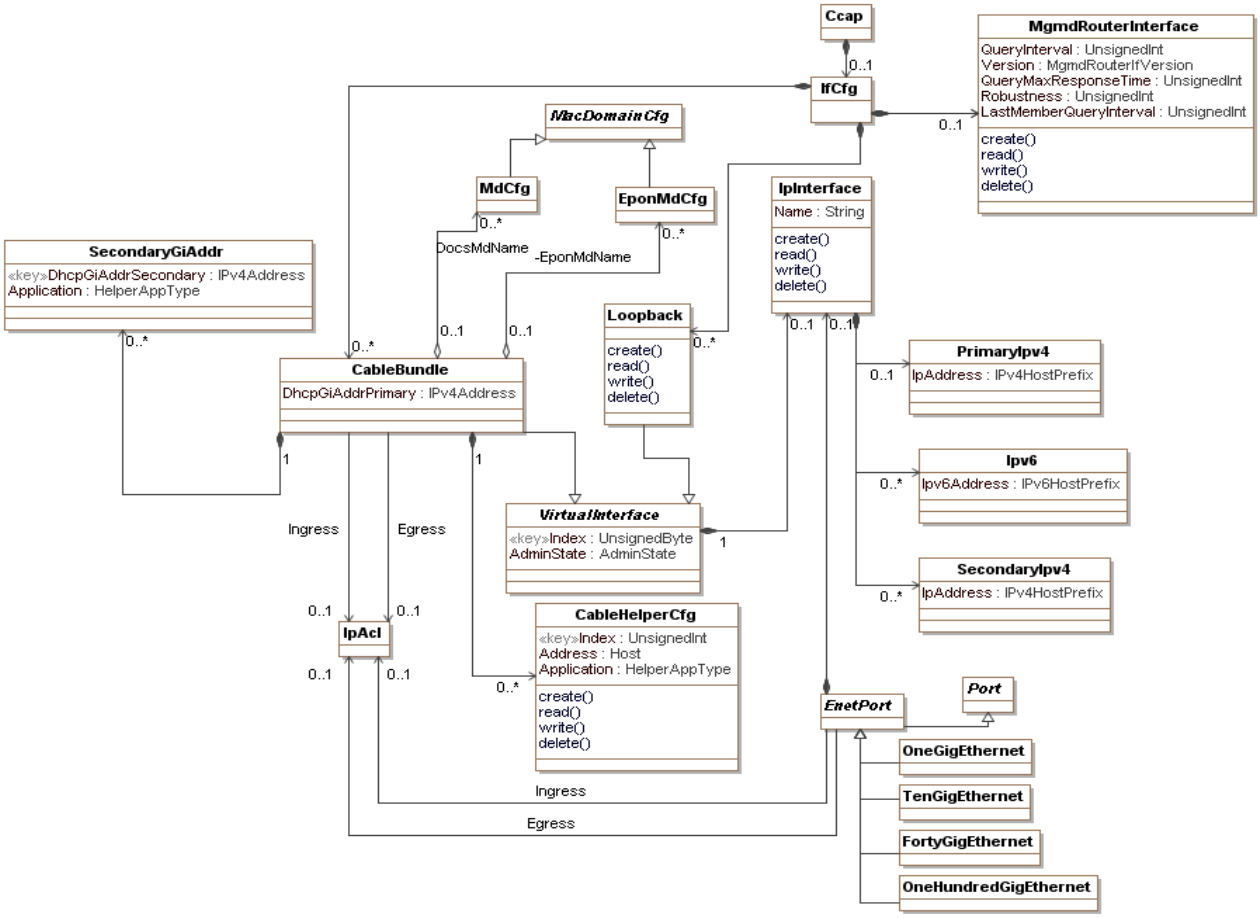


Figure 6.18: Interface Configuration Objects

6.5.9.1 Ccap

This configuration object is included in figure 6.18 for reference. It is defined in clause 6.5.4.1.

6.5.9.2 IfCfg

The IfCfg object is the primary container of interface configuration objects.

Table 6.207: IfCfg Object Associations

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

6.5.9.3 Loopback

A loopback interface is a logical interface that is not tied to a specific hardware port. The CCAP shall support a loopback interface to provide a virtual interface to assist in overall system configuration.

Table 6.208: Loopback Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
VirtualInterface	Specialization of VirtualInterface			

6.5.9.4 VirtualInterface

The VirtualInterface abstract object contains attributes shared by CCAP virtual interfaces (Loopback and CableBundle).

Table 6.209: VirtualInterfaceObject Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedByte	Yes (Key)			
AdminState	AdminState	No			down

Table 6.210: VirtualInterface Object Associations

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

6.5.9.5 IpInterface

IpInterface is an object used to configure an IP interface on the CCAP. Attributes from this object are used by the CableBundle, Loopback, and EnetPort objects. For a CCAP operating in non-routing mode, an IpInterface instance need not be configured for CableBundle objects.

Table 6.211: IpInterface Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Name	String	Yes (Key)			

IpInterface has several associations.

Table 6.212: IpInterface Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
PrimaryIpv4	Directed composition to PrimaryIpv4		0..1	
Ipv6	Directed composition to Ipv6		0..*	
SecondaryIpv4	Directed composition to SecondaryIpv4		0..*	

6.5.9.5.1 IpInterface Object Attributes

6.5.9.5.1.1 Name

The name for this instance of an interface. This name is used to reference a specific IpInterface instance and associate it with the referring object.

6.5.9.6 PrimaryIpv4

The PrimaryIpv4 object allows a primary IPv4 interface address to be configured.

Table 6.213: PrimaryIpv4 Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
IpAddress	Ipv4HostPrefix	Yes			

6.5.9.6.1 PrimaryIpv4 Attributes

6.5.9.6.1.1 IpAddress

This attribute configures the IPv4 address and prefix for this instance.

6.5.9.7 Ipv6

The PrimaryIpv6 object allows a primary IPv6 interface address to be configured. For IPv6 addresses, the concept of primary and secondary does not apply; for this reason, a list of IPv6 addresses may be configured.

Table 6.214: Ipv6 Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Ipv6Address	Ipv6HostPrefix	Yes			

6.5.9.7.1 Ipv6 Attributes

6.5.9.7.1.1 Ipv6Address

This attribute configures the IPv6 address and prefix for this instance.

6.5.9.8 SecondaryIpv4

The SecondaryIpv4 object allows secondary IPv4 addresses to be configured.

Table 6.215: SecondaryIpv4 Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
IpAddress	Ipv4HostPrefix	Yes			

6.5.9.8.1 SecondaryIpv4 Attributes

6.5.9.8.1.1 IpAddress

This attribute configures the IPv4 address and prefix for this instance.

6.5.9.9 CableBundle

A CableBundle is a compact way of assigning Layer 3 network addresses to a set of Layer 2 interfaces. This allows the bundled Layer 2 interfaces to share a common pool of IPv4 Subnets or IPv6 prefixes so that these IP address resources can be efficiently used by the CCAP operating in routing mode.

Table 6.216: CableBundle Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
DhcpGiAddrPrimary	Ipv4Address	Yes			

A CableBundle can only be associated with MAC domains of a given type; the CCAP shall reject the configuration of a CableBundle instance in which both an MdCfg and an EponMdCfg have been configured.

Table 6.217: CableBundle Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
MdCfg	Directed aggregation to MdCfg	0..1	0..*	DocsMdName
EponMdCfg	Directed aggregation to EponMdCfg	0..1	0..*	EponMdName
CableHelperCfg	Directed composition to CableHelperCfg	1	0..*	
SecondaryGiAddr	Directed composition to SecondaryGiAddr	1	0..*	
IpAcl	Directed association to IpAcl		0..1	Ingress
IpAcl	Directed association to IpAcl		0..1	Egress

6.5.9.9.1 CableBundle Object Attributes

6.5.9.9.1.1 DhcpGiAddrPrimary

This attribute configures how the DHCP relay agent populates the GiAddr for relayed DHCP traffic on the CCAP in routing mode.

6.5.9.10 CableHelperCfg

The CableHelperCfg configuration object allows the operator to configure different Cable Helper addresses for DHCP Clients. The CCAP operating in routing mode ties these Cable Helper addresses to the CableBundle interfaces and the MAC Domains they service.

Table 6.218: CableHelperCfg Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes (Key)			
Address	Host	Yes			
Application	Enum	No	other(1) host(2) mta(3) stb(4) cm(5) all(6)		all

6.5.9.10.1 CableHelperCfg Object Attributes

6.5.9.10.1.1 Index

The index for the CableHelperCfg instance.

6.5.9.10.1.2 Address

This attribute configures the IP address or FQDN of the DHCP server configured as a cable helper.

6.5.9.10.1.3 Application

This attribute configures the device class for which this cable helper configuration applies. The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.9.11 SecondaryGiAddr

This object allows a secondary GiAddr to be configured for a CableBundle instance.

Table 6.219: SecondaryGiAddr Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
DhcpGiAddrSecondary	Ipv4Address	Yes (Key)			
Application	Enum	No	other(1) host(2) mta(3) stb(4) cm(5) all(6)		all

6.5.9.11.1 SecondaryGiAddr Object Attributes

6.5.9.11.1.1 DhcpGiAddrSecondary

This attribute configures how the DHCP relay agent populates the secondary GiAddr for relayed DHCP traffic on the CCAP in routing mode.

6.5.9.11.1.2 Application

This attribute configures the device class for which this GiAddr instance applies. The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.9.12 MacDomainCfg

This configuration object is included in figure 6.18 for reference. It is defined in clause 6.5.7.6.6.

6.5.9.13 EponMdCfg

This configuration object is included in figure 6.18 for reference. It is defined in clause 6.5.11.6.

6.5.9.14 MdCfg

This configuration object is included in figure 6.18 for reference. It is defined in clause 6.5.7.6.4.

6.5.9.15 EnetPort

This configuration object is included in figure 6.18 for reference. It is defined in clause 6.5.5.21.

6.5.9.16 OneGigEthernet

This configuration object is included in figure 6.18 for reference. It is defined in clause 6.5.5.22.

6.5.9.17 TenGigEthernet

This configuration object is included in figure 6.18 for reference. It is defined in clause 6.5.5.23.

6.5.9.18 FortyGigEthernet

This configuration object is included in figure 6.18 for reference. It is defined in clause 6.5.5.24.

6.5.9.19 OneHundredGigEthernet

This configuration object is included in figure 6.18 for reference. It is defined in clause 6.5.5.25.

6.5.9.20 Port

This configuration object is included in figure 6.18 for reference. It is defined in clause 6.5.5.10.

6.5.9.21 MgmdRouterInterface

This configuration object allows for configuration of the CCAP IP Multicast Router. These configuration objects are defined in the Multicast Group Membership Discovery MIB, [i.17]. Table 6.220 is derived from this MIB.

Table 6.220: MgmdRouterInterface Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
QueryInterval	UnsignedInt	No		seconds	125
Version	Enum	No	other(1), igmpv1(2), igmpv2OrMldv1(3), igmpv3OrMldv2(4)		igmpv2OrMldv1
QueryMaxResponseTime	UnsignedInt	No	0..31 744	tenths of seconds	100
Robustness	UnsignedInt	No	1..225		2
LastMemberQueryInterval	UnsignedInt	No	0..31 744	tenths of seconds	10

6.5.9.21.1 MgmdRouterInterface Object Attributes

6.5.9.21.1.1 QueryInterval

The frequency in seconds at which IGMP or MLD Host-Query packets are transmitted on this interface.

6.5.9.21.1.2 Version

The version of MGMD that is running on this interface. Value 2 applies to IGMPv1 routers only. Value 3 applies to IGMPv2 and MLDv1 routers, and value 4 applies to IGMPv3 and MLDv2 routers.

This object can be used to configure a router capable of running either version. For IGMP and MLD to function correctly, all routers on a LAN need to be configured to run the same version on that LAN.

6.5.9.21.1.3 QueryMaxResponseTime

The maximum query response interval in seconds advertised in MGMDv2 or IGMPv3 queries on this interface.

6.5.9.21.1.4 Robustness

The robustness variable utilized by an MGMDv3 host in sending state-change reports for multicast routers. To ensure the state-change report is not missed, the host retransmits the state-change report [mgmdHostInterfaceVersion3Robustness - 1] times. The variable needs to be a non-zero value.

6.5.9.21.1.5 LastMemberQueryInterval

The Last Member Query Interval is the Max Query Response Interval in tenths of a second inserted into group-specific queries sent in response to leave group messages, and is also the amount of time between group-specific query messages. This value may be tuned to modify the leave latency of the network. A reduced value results in reduced time to detect the loss of the last member of a group. The value of this object is irrelevant if mgmdRouterInterfaceVersion is 1.

6.5.10 Management Configuration Objects

The management configuration objects configure fault management and SNMP for the CCAP.

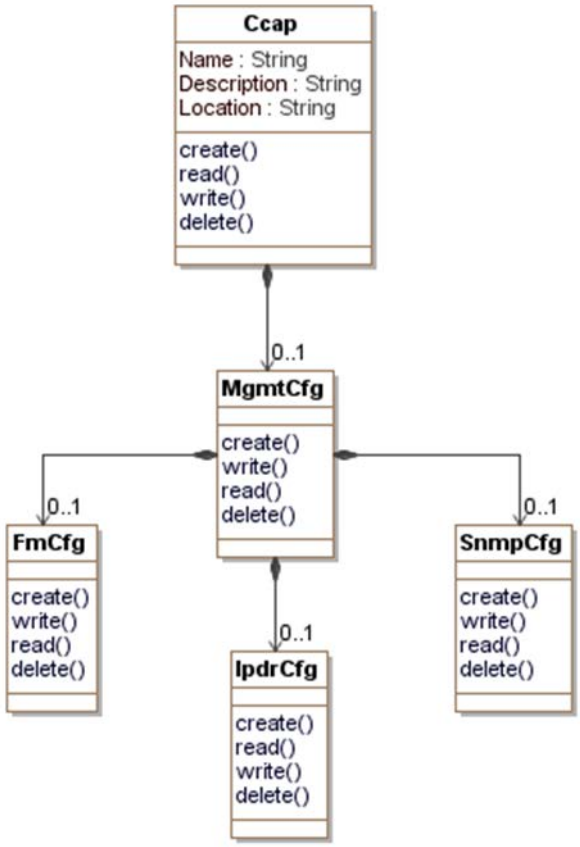


Figure 6.19: Management Configuration Objects

6.5.10.1 Ccap

This configuration object is included in figure 6.19 for reference. It is defined in clause 6.5.4.1.

6.5.10.2 MgmtCfg

The MgmtCfg object is the primary container of the management configuration objects. It has the following associations:

Table 6.221: MgmtCfg Object Associations

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

6.5.10.3 FmCfg

This configuration object is included in figure 6.19 for reference. It is defined in clause 6.5.10.6.2.

6.5.10.4 SnmpCfg

This configuration object is included in figure 6.19 for reference. It is defined in clause 6.5.10.7.2.

6.5.10.5 IpdrCfg

This configuration object is included in figure 6.19 for reference. It is defined in clause 6.5.10.8.2.

6.5.10.6 Fault Management Configuration Objects

The CCAP will employ much of the event reporting methods that have long been a part of DOCSIS and PMI. This section will detail the configuration portions of the event reporting infrastructure which have been adapted from [7]. The Object model for these configured objects is shown in figure 6.20.

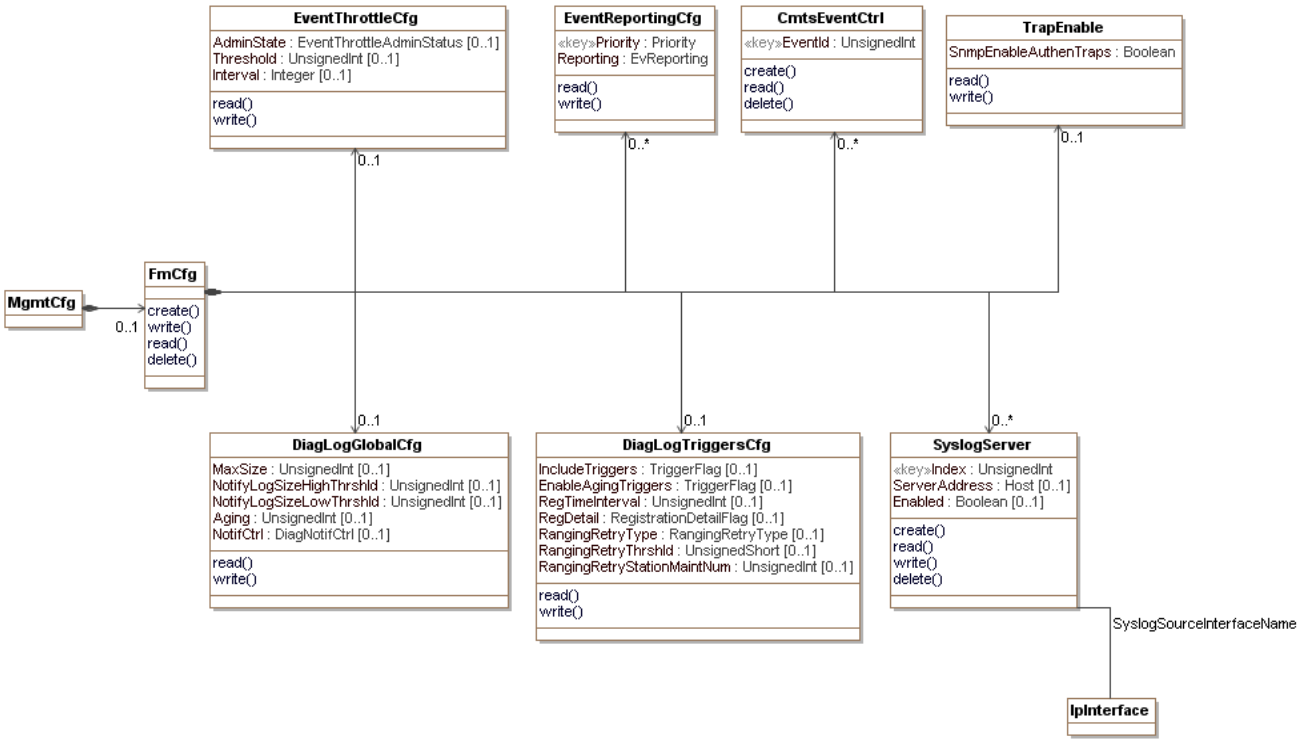


Figure 6.20: Fault Management Configuration Objects

These objects allow the operator to configure logging for various events so these issues can be tracked.

6.5.10.6.1 MgmtCfg

This configuration object is included in figure 6.19 for reference. It is defined in clause 6.5.10.2.

6.5.10.6.2 FmCfg

The FmCfg object is the primary container of fault management configuration objects. It has the following associations:

Table 6.222: FmCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
EventThrottleCfg	Directed composition to EventThrottleCfg		0..1	
EventReportingCfg	Directed composition to EventReportingCfg		0..*	
CmtsEventCtrl	Directed composition to CmtsEventCtrl		0..*	
TrapEnable	Directed composition to TrapEnable		0..1	
DiagLogGlobalCfg	Directed composition to DiagLogGlobalCfg		0..1	
DiagLogTriggersCfg	Directed composition to DiagLogTriggersCfg		0..1	
SyslogServer	Directed composition to SyslogServer		0..*	

6.5.10.6.3 EventThrottleCfg

This configuration object is based on the docsDevEvent group defined in RFC 4639 [21] and uses the following attributes without modification for CCAP:

- AdminStatus (renamed AdminState)
- Threshold
- Interval

Reference: RFC 4639 [21], docsDevEvent Group.

6.5.10.6.4 EventReportingCfg

This configuration object is based on the docsDevEvControlTable object defined in RFC 4639 [21] and will be used without modification for CCAP.

Reference: RFC 4639 [21], docsDevEvControlTable.

6.5.10.6.5 CmtsEventCtrl

This configuration object is defined in [7] and will be used with no modifications for CCAP.

This object represents the control mechanism to enable the dispatching of events based on the Event Id. The following rules define the event control behavior:

- If the CmtsEventCtrl object has no instances or contains an instance with Event ID 0, then all events matching the Local Log settings of docsDevEvReporting are sent to local log ONLY.
- Additionally, if The CmtsEventCtrl object contains configured instances, then Events matching the Event Ids configured in the object are sent according to the settings of the docsDevEvReporting object; i.e. Traps, Syslog, etc.

Reference: [7], CmtsEventCtrl Object section.

6.5.10.6.6 TrapEnable

This configuration object contains attributes which allow enabling or disabling of SNMP Notifications. The SnmpEnableAuthenTraps attribute is taken from RFC 3418 [13] and will be used without modification for the CCAP.

Reference: RFC 3418 [13], snmpEnableAuthenTraps.

6.5.10.6.7 DiagLogGlobalCfg

This configuration object is based on the LogGlobal object defined in [7] and used with modifications for the CCAP. The following read-only attributes have been removed:

- CurrentSize
- LastResetTime
- LastClearTime

The following attributes have been moved to the DiagLogGloblalCtrl object:

- ResetAll
- ClearAll

This object defines the parameters to manage and control the instantiation of CMs in the Diagnostic Log object.

Reference: [7], LogGlobal Object section.

6.5.10.6.8 DiagLogTriggersCfg

This configuration object is based on the LogTriggersCfg object in Annex G of [7] and will be used without modification for CCAP.

This object defines the parameters to configure the Diagnostic Log triggers. One or more triggers can be configured to define the actions of creating or updating CM entries into the Diagnostic Log.

Reference: [7], LogTriggersCfg Object section.

6.5.10.6.9 SyslogServer

This object allows the configuration of a specific Syslog Server.

Table 6.223: SyslogServer Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Index	UnsignedInt	Yes (Key)			
ServerAddress	Host	Yes			
Enabled	Boolean	No			false

Table 6.224: SyslogServer Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
IpInterface	Directed composition to IpInterface			SyslogSourceInterfaceName

6.5.10.6.9.1 SyslogServer Object Attributes

6.5.10.6.9.1.1 Index

This key represents the unique identifier of an instance in this object.

6.5.10.6.9.1.2 ServerAddress

This attribute represents the IP address of the Syslog server. If DNS is supported, this attribute can contain the FQDN of the Syslog server.

6.5.10.6.9.1.3 Enabled

Indicates if the Syslog server is used for sending Syslog messages or is disabled.

6.5.10.7 SNMP Agent Configuration Objects

The configuration objects for the CCAP SNMP Agent shown in figure 6.21. This is only a policy configuration, but can be matched to full SNMPv3 implementations using similar procedures as done for TLV 38, 53, and 54 described in [7].

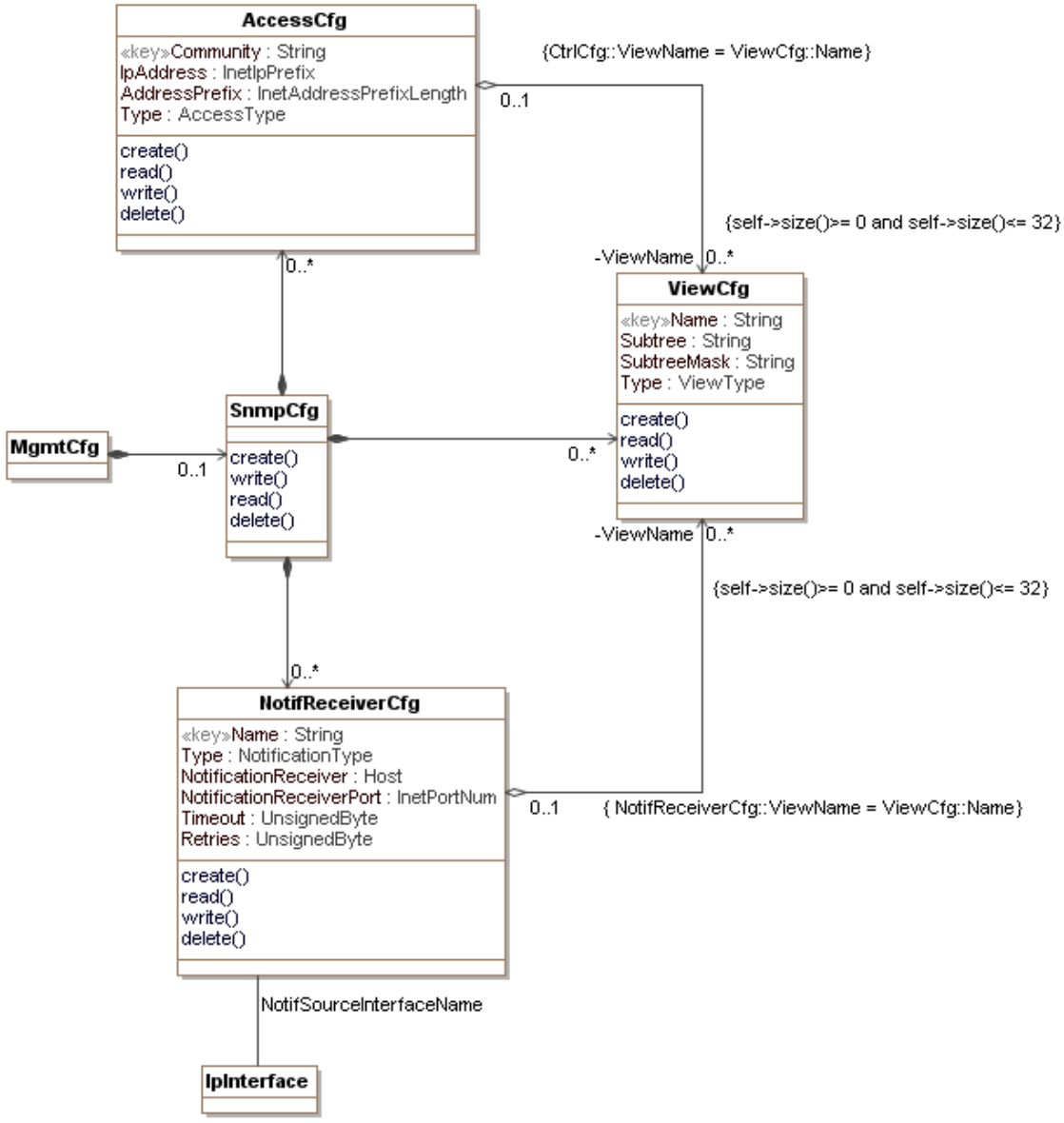


Figure 6.21: SNMP Agent Configuration Objects

6.5.10.7.1 MgmtCfg

This configuration object is included in figure 6.19 for reference. It is defined in clause 6.5.10.2.

6.5.10.7.2 SnmpCfg

The SnmpCfg object is the primary container of SNMP configuration objects. It has the following associations:

Table 6.225: FmCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
AccessCfg	Directed composition to AccessCfg		0..*	
ViewCfg	Directed composition to ViewCfg		0..*	
NotifReceiverCfg	Directed composition to NotifReceiverCfg		0..*	

6.5.10.7.3 AccessCfg

This object defines the configuration of access control for SNMPv1/v2c received request messages. When a SNMP request message is received, the system checks the validity of the request by matching the community string, source (IP address, subnet), access type and view restrictions for included SNMP OIDs in the request.

Table 6.226: AccessCfg Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Community	String	Yes (Key)	1..32		
IpAddress	InetIpPrefix	Yes			
AddressPrefix	InetAddressPrefixLength	Yes			
Type	Enum	No	readOnly(1), readWrite(2)		readOnly

Table 6.227: AccessCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
ViewCfg	Directed aggregation to ViewCfg	0..1	0..*	ViewName

6.5.10.7.3.1 AccessCfg Object Attributes

6.5.10.7.3.1.1 Community

The community string defined for the access control rule.

6.5.10.7.3.1.2 IpAddress

The address used in conjunction with the AddressPrefix attribute used to validate the source of an incoming SNMP request.

6.5.10.7.3.1.3 AddressPrefix

The prefix to apply to the IpAddress attribute for matching valid sources for the SNMP requests.

6.5.10.7.3.1.4 Type

Defines the type of access granted to the SNMP request. An enumeration of "other" was purposefully excluded from this enumeration.

6.5.10.7.4 ViewCfg

This object defines a View consisting of a single OID subtree matching rule for inclusion or exclusion as part of a SNMP message processing procedure such as access authorization or dispatch or notifications.

Table 6.228: ViewCfg Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Name	String	Yes (Key)			
Subtree	String	Yes			
SubtreeMask	String	Yes			
Type	Enum	Yes	other(1), included(2), excluded(3)		

6.5.10.7.4.1 ViewCfg Object Attributes

6.5.10.7.4.1.1 Name

The administrative name of an instance of this object.

6.5.10.7.4.1.2 Subtree

The OID subtree to be matched for the access view. This attribute is formatted as the text representation of an ASN.1 OID following the ABNF notation below:

Subtree = empty | OID [.OID]*

OID = number; 0..128

The matching procedures are borrowed from RFC 3414 [i.11] for tree views matching with the difference that the configuration elements uses a text notation to represent OIDs and OID masks. See the SubtreeMask attribute definition for further information.

6.5.10.7.4.1.3 SubtreeMask

A mask to match OIDs for inclusion or exclusion as part of the view. This attribute definition is borrowed from RFC 3414 [i.11]. The only difference is that instead of bits per OID, a byte of value 0 or 1 is used to represent this attribute.

Each byte value 1 indicates the inclusion of the corresponding OID position in the Subtree attribute, while the value 0 indicates no need to match. See RFC 3414 [i.11] for details.

6.5.10.7.4.1.4 Type

Indicates inclusion or exclusion of the subtree for the defined view.

6.5.10.7.5 NotifReceiverCfg

This object defines where to send notifications. When an event is to be dispatched as a notification, the system checks for instances of this object that have the notification OID associated with the event as part of their Inclusion list in their ViewCfg instances. The system then sends notifications based on the matched occurrences per their configured parameters.

If an instance of NotifSourceInterfaceName is not configured, then selection of notification source interface is vendor proprietary.

Table 6.229: NotifReceiverCfg Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Name	String	Yes (Key)	1..32		
Type	Enum	No	snmpV1Trap(1), snmpV2cTrap(2), snmpV2Inform(3)		snmpV2cTrap
NotificationReceiver	Host	Yes			
NotificationReceiverPort	InetPortNum	No			162
Timeout	UnsignedByte	No		seconds	1
Retries	UnsignedByte	No			3

Table 6.230: NotifReceiverCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
ViewCfg	Directed aggregation to ViewCfg	0..1	0..*	ViewName
IpInterface	Association with IpInterface			NotifSourceInterfaceName

6.5.10.7.5.1 NotifReceiverCfg Object Attributes

6.5.10.7.5.1.1 Name

The administrative name of an instance in this object.

6.5.10.7.5.1.2 Type

Indicates the type of SNMP notification being sent:

- snmpV1Trap: SNMP v1 trap
- snmpV2cTrap: SNMP v2c trap
- snmpV2cInform: SNMP v2c Inform

An enumeration of "other" was purposefully excluded from this enumeration.

6.5.10.7.5.1.3 NotificationReceiver

The IP address or FQDN of the notification receiver.

6.5.10.7.5.1.4 Port

The UDP port the notification receiver listen for messages.

6.5.10.7.5.1.5 Timeout

The time in seconds the sender waits for receiving confirmation for a notification being sent. This attribute is meaningful only when the attribute Type is set to snmpV2cInform(4); otherwise it is ignored.

6.5.10.7.5.1.6 Retries

The number of retries the sender will attempt in case of it has not received confirmation of inform reception. This attribute is meaningful only when the attribute Type is set to snmpV2cInform(4); otherwise it is ignored.

6.5.10.8 IPDR Configuration Objects

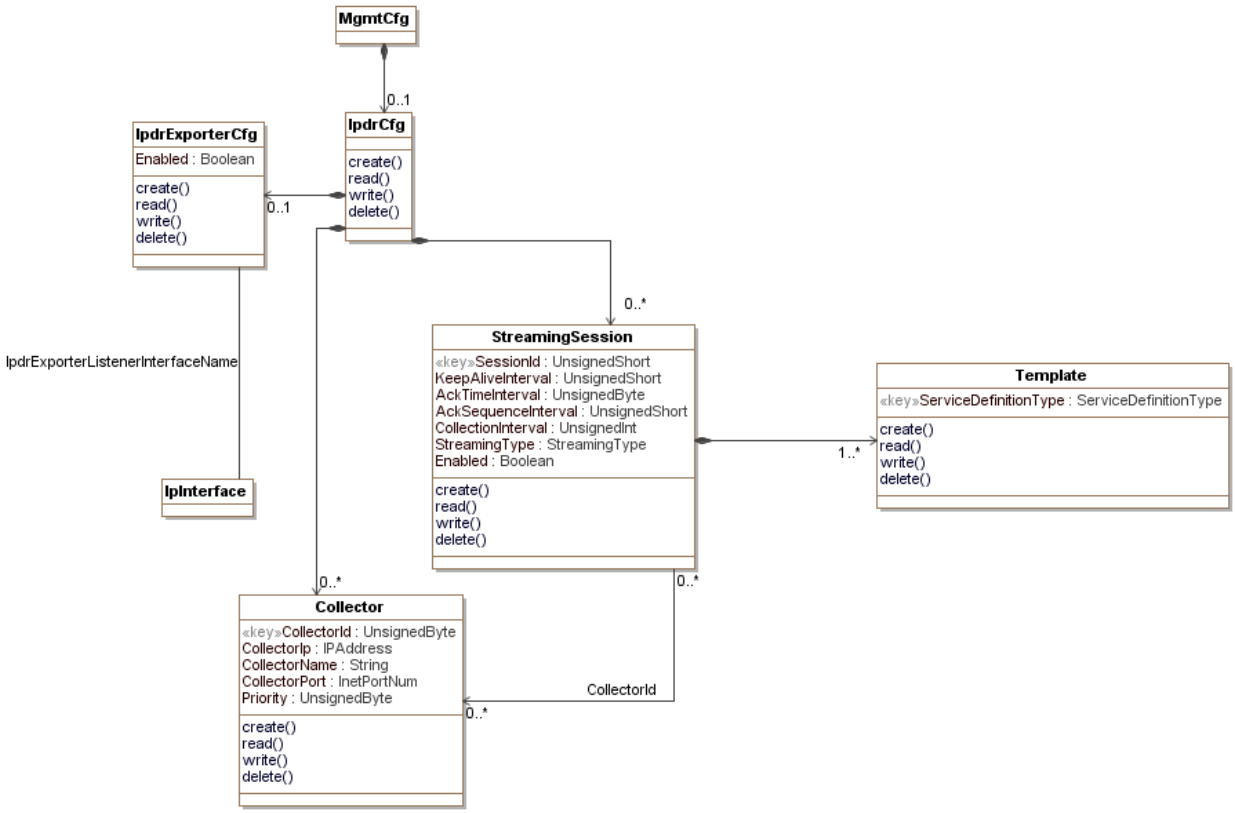


Figure 6.22: IPDR Configuration Objects

6.5.10.8.1 MgmtCfg

This configuration object is included in figure 6.19 for reference. It is defined in clause 6.5.10.2.

6.5.10.8.2 IpdrCfg

The IpdrCfg object is the container for the Internet Protocol Detail Records (IPDR) configuration objects. It has the following associations:

Table 6.231: IpdrCfg Object Associations

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

6.5.10.8.3 IpdrExporterCfg

This configuration object allows an exporter to be turned on and off.

Table 6.232: IpdrExporterCfg Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
Enabled	Boolean	No			true

Table 6.233: IpdrExporterCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
IpInterface	Association with IpInterface			IpdrExporterListenerInterfaceName

When an IP interface is selected, this specifies the IP interface on which the IPDR server listens. If an IP interface is not specified, the behavior of the CCAP is vendor specific.

6.5.10.8.3.1 IpdrExporterCfg Object Attributes

6.5.10.8.3.1.2 Enabled

This attribute configures whether or not the IPDR exporter is enabled.

6.5.10.8.4 StreamingSession

This configuration object is used to configure global IPDR connection attributes. A typical use case is for a single Template to be associated with a StreamingSession.

Table 6.234: StreamingSession Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
SessionId	UnsignedShort	Yes (Key)			
KeepAliveInterval	UnsignedShort	No		seconds	20
AckTimeInterval	UnsignedByte	No	1..60	seconds	30
AckSequenceInterval	UnsignedShort	No	1..500	records	200
CollectionInterval	UnsignedInt	Yes	0..86 400	seconds	
StreamingType	Enum	Yes	other(1), timeInterval(2), adHoc(3), event(4), timeEvent(5)		
Enabled	Boolean	No			true

Table 6.235: StreamingSession Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
Template	Directed composition to Template		1..*	
Collector	Directed association to Collector	0..*	0..*	CollectorId

6.5.10.8.4.1 StreamingSession Object Attributes

6.5.10.8.4.1.1 SessionId

This attribute configures the ID for this session instance.

6.5.10.8.4.1.2 KeepAliveInterval

This attribute configures the interval in seconds at which IPDR "keepalives" are sent from the CCAP IPDR exporter to the collector.

6.5.10.8.4.1.3 AckTimeInterval

This attribute configures the interval in seconds in which the CCAP IPDR exporter waits for an acknowledgment.

6.5.10.8.4.1.4 AckSequenceInterval

This attribute configures the maximum number of unacknowledged records that can be sent by the CCAP IPDR exporter before receiving an acknowledgement.

6.5.10.8.4.1.5 CollectionInterval

Where streaming is of the type timeInterval, this attribute configures the interval in seconds at which IPDR information is extracted from the CCAP management objects and transmitted to the collector.

Where streaming is of the type timeEvent, this attribute identifies the interval at which the CCAP IPDR exporter will close the IPDR session to allow IPDR session processing to occur. Records created by Service Definitions supporting timeEvent are sent when the event is generated.

6.5.10.8.4.1.6 StreamingType

This attribute configures the type of IPDR streaming used for the session. See the IPDR Service Definition Schemas section of [7] for the streaming types supported by each Service Definition. The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.10.8.4.1.7 Enabled

This attribute controls whether the IPDR Session is enabled or disabled.

6.5.10.8.5 Template

This configuration object allows the configuration of an individual IPDR session for a given IPDR connection.

Table 6.236: Template Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
ServiceDefinitionType	Enum	Yes (Key)	other(1), cmtsCmServiceFlowType(2), cmtsCmRegStatusType(3), cmtsCmUsStatsType(4), cmtsDsUtilStatsType(5), cmtsUsUtilStatsType(6), cmtsTopologyType(7), cpeType(8), diagLogType(9), diagLogDetailType(10), diagLogEventType(11), samisType1(12), samisType2(13), spectrumMeasurementType(14)		

6.5.10.8.5.1 Template Object Attributes

6.5.10.8.5.1.1 ServiceDefinitionType

This attribute configures the service type definition for this IPDR session. See the IPDR Service Definition Schemas section of [7] for the definitions and schemas of the types defined in this enumeration. The value of other(1) is used when a vendor-extension has been implemented for this attribute.

6.5.10.8.6 Collector

This configuration object allows the operator to configure an IPDR collector.

Table 6.237: Collector Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
CollectorId	UnsignedByte	Yes (Key)			
CollectorIp	IpAddress	Yes			
CollectorName	String	No			""
CollectorPort	InetPortNum	No			4 737
Priority	UnsignedByte	Yes			

6.5.10.8.6.1 Collector Object Attributes

6.5.10.8.61.1 CollectorId

This key configures a unique identifier for this collector instance.

6.5.10.8.6.1.2 CollectorIp

This attribute configures the IP address of collectors from which the CCAP will accept a connect. As per [7], the collector establishes a connection to the CCAP.

6.5.10.8.6.1.3 CollectorName

This attribute configures a name for the IPDR collector.

6.5.10.8.6.1.4 CollectorPort

This attribute configures the port used by the collector to communicate with the CCAP. The default for this is 4737.

6.5.10.8.6.1.5 Priority

This attribute configures the priority of this IPDR collector. The priority is used to elect the primary and active collector. The collector with the lowest priority is elected.

6.5.11 CCAP EPON Configuration Objects

For DOCSIS EPON provisioning and management, the CCAP shall meet the requirements in the [1]. The EPON configuration objects are shown in the following diagram.

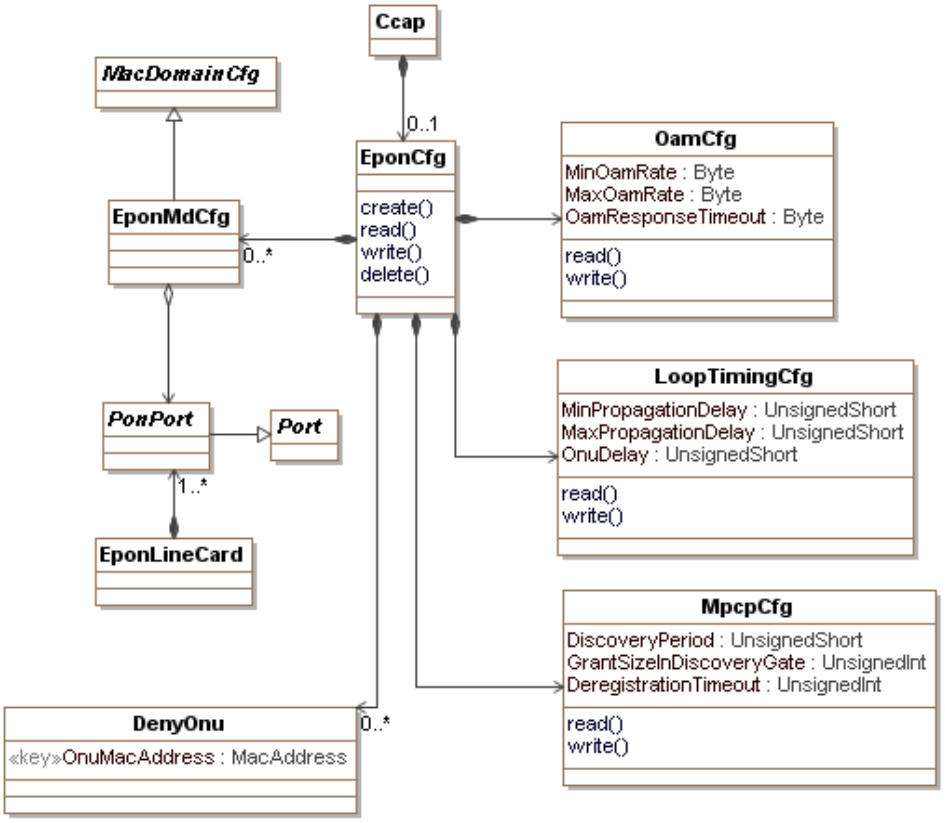


Figure 6.23: EPON Configuration Objects

6.5.11.1 Ccap

This configuration object is included in figure 6.23 for reference. It is defined in clause 6.5.4.1.

6.5.11.2 EponCfg

The EponCfg object is the primary container of EPON configuration objects. It has the following associations:

Table 6.238: EponCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
EponMdCfg	Directed composition to EponMdCfg		0..*	
OamCfg	Directed composition to OamCfg			
LoopTimingCfg	Directed composition to LoopTimingCfg			
MpcpCfg	Directed composition to MpcpCfg			
DenyOnu	Directed composition to DenyOnu		0..*	

6.5.11.3 OamCfg

This configuration object is taken from [1] and is used without modification for CCAP. This object controls the rate at which OAM messages are sent on the EPON interface.

Reference: [1], EPON OAM Configuration section.

6.5.11.4 LoopTimingCfg

This configuration object is taken from [1] and is used with the following modifications for CCAP: the OltUpDownDelayOffset and NullGrantSize attributes have been removed.

This object configures the loop timing for EPON interfaces.

Reference: [1], Loop Timing section.

6.5.11.5 MpcpCfg

This configuration object is taken from [1] and is used without modification for CCAP. It configures the Multi-Point Control Protocol for EPON interfaces.

Reference: [1], MPCP Configuration section.

6.5.11.6 EponMdCfg

This object defines a specialization of the MacDomain object for EPON interfaces.

Table 6.239: EponMdCfg Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
MacDomainConfig	Specialization of MacDomainCfg			
PonPort	Directed aggregation to PonPort			

6.5.11.7 DenyOnu

This configuration object allows an operator to create a list of ONU MAC addresses that are not allowed to register.

Table 6.240: DenyOnu Object Attributes

Attribute Name	Type	Required Attribute	Type Constraints	Units	Default Value
OnuMacAddress	MacAddress	Yes (Key)			

6.5.11.7.1 DenyOnu Object Attributes

6.5.11.7.1.1 OnuMacAddress

The MAC address of the ONU that will be added to the deny list. This attribute is used as a key.

6.5.11.8 MacDomainCfg

This configuration object is included in figure 6.23 for reference. It is defined in clause 6.5.7.6.6.

6.5.11.9 PonPort

This configuration object is included in figure 6.23 for reference. It is defined in clause 6.5.5.26.

6.5.11.10 Port

This configuration object is included in figure 6.23 for reference. It is defined in clause 6.5.5.10.

6.5.11.11 EponLineCard

This configuration object is included in figure 6.23 for reference. It is defined in clause 6.5.5.7.

6.6 Status Monitoring and Control Requirements

6.6.1 Status Monitoring and Control UML Object Models

This section defines the object models for the utilization of CCAP status and control management functions. These objects are typically not used during installation when the CCAP is brought on-line and into service. Status and control management objects are used at run time to obtain status information or command actionable control. Examples of control functions include clearing an event log or starting a packet capture on a specific MAC Domain. Examples of status functions include checking the operational state of an interface or the results of a diagnostics test. In general, configuration of these control objects would not be included in the startup-config for initial CCAP device configuration.

6.6.1.1 Fault Management Control Objects

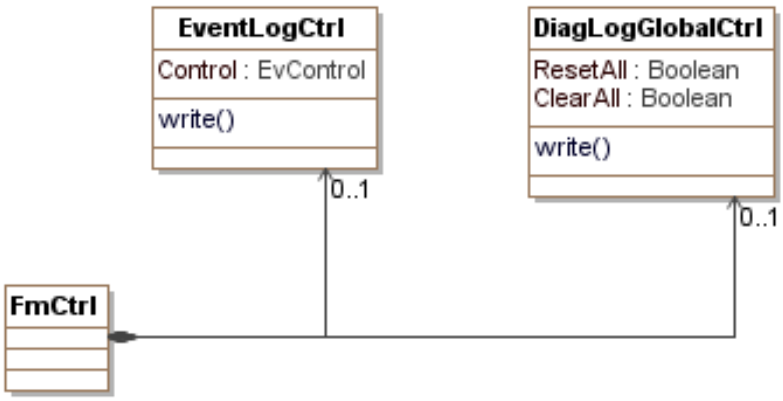


Figure 6.24: Fault Management Control Objects

6.6.1.1.1 FmCtrl

The FmCtrl object is the primary container of Fault Management Control objects. It has the following associations:

Table 6.241: FmCtrl Object Associations

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

6.6.1.1.2 EventLogCtrl

This control object is based on the docsDevEvent group defined in [21] and contains a single actionable configuration attribute: Control. This object is used to clear the event log or to return all event priorities to their default settings.

Reference: [21], docsDevEvControl object.

6.6.1.1.3 DiagLogGlobalCtrl

This control object is based on the LogGlobal object defined in [7] and contains the following actionable configuration attributes:

- ResetAll
- ClearAll

This object allows Log and LogDetail instances to be reset or cleared.

Reference: [7], LogGlobal Object section.

6.6.1.2 Performance Management Control Objects

The objects in the Performance Management Control class diagram are taken from the following DOCSIS MIBs and are used without modification for the CCAP:

Table 6.242: Performance Management Control Objects

Object	MIB
SignalQualityExt	DOCS-IF3-MIB
CmtsSpectrumAnalysisMeas	DOCS-IF3-MIB
CmtsSignalQualityExt	DOCS-IF3-MIB
CmtsCmCtrlCmd	DOCS-IF3-MIB
CmtsDebugDsid	DOCS-QOS3-MIB
CmtsDebugDsidStats	DOCS-QOS3-MIB
ChgOverGroup	DOCS-LOADBAL3-MIB
ChgOverStatus	DOCS-LOADBAL3-MIB

Reference: [7], DOCS-IF3-MIB, DOCS-QOS3-MIB, DOCS-LOADBAL3-MIB.

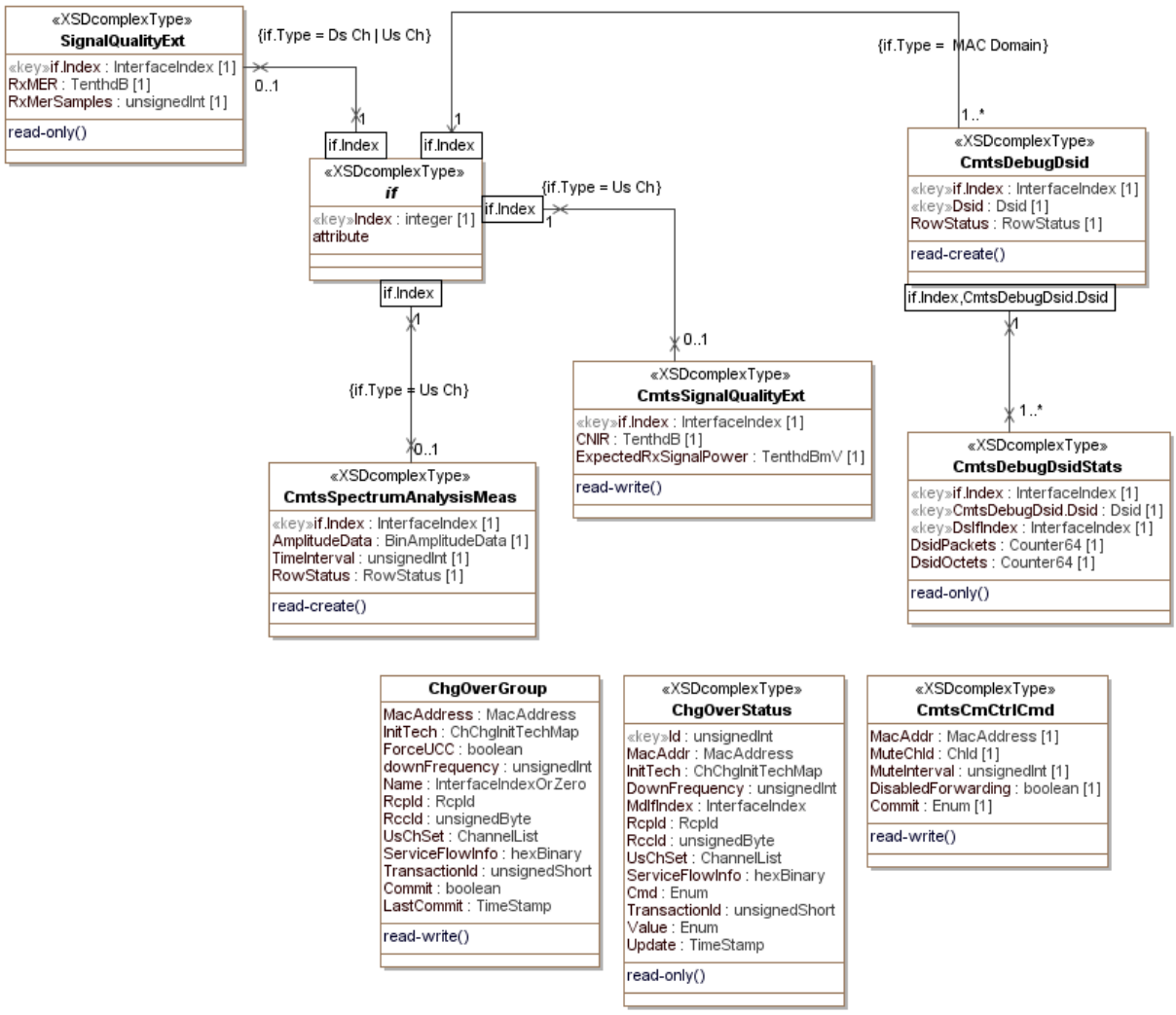


Figure 6.25: Performance Management Control Objects

7 Performance Management

7.1 Performance Management Requirements and Transport Protocols

7.1.1 SNMP and MIB Requirements

The SNMP requirements of the CCAP are based upon the requirements specified in [7] and to a lesser extent [i.5]. The SNMP requirements in this section supersede those requirements, except where noted.

Since CCAP configuration will be primarily accomplished via the standard XML configuration file and legacy CLI commands, SNMP is not used as a primary configuration interface on the CCAP. Based on this, most CCAP MIB objects will be used in a read-only mode for status and performance monitoring.

In addition, the CCAP requires a very small set of read-create or read-write MIB objects used by operators for specific automation tasks that would be too cumbersome to execute via CLI or Configuration File. These read-create or read-write objects are typically part of vendor-proprietary MIBs not mentioned in the present document.

7.1.1.1 Protocol and Agent Requirements

The CCAP shall support SNMPv1 and SNMPv2.

The CCAP may support SNMPv3.

The CCAP shall support at least 10 SNMP Community strings with controlled access via access lists.

7.1.1.2 MIBs

The CCAP shall support read-only access for all CMTS Mandatory ("M") MIB objects that have an SNMP access type of read-only ("RO") in Annex A of [7] and Annex A of ES 203 085 [3].

The CCAP shall support read-only access for all CMTS Mandatory ("M") MIB objects that have an SNMP access type of read-write ("RW") or read-create ("RC") in Annex A of [7] and Annex A of ES 203 085 [3].

The CCAP may support read-write access for all CMTS Mandatory ("M") MIB objects that have an SNMP access type of read-write ("RW") in Annex A of [7] and Annex A of ES 203 085 [3].

The CCAP may support read-create access for all CMTS Mandatory ("M") MIB objects that have an SNMP access type of read-create ("RC") in Annex A of [7] and Annex A of ES 203 085 [3].

7.1.1.3 SCTE MIBs

The CCAP shall support all mandatory MIB objects specified in the tables in Annex F, Detailed MIB Requirements.

For video sessions created via static configuration (e.g. via XML configuration file), the CCAP shall instantiate the appropriate row entries in the SCTE-HMS-MPEG-MIB's mpegProgramMappingTable, mpegVideoSessionTable, mpegVideoSessionPtrTable, and mpegInputTSOutputSessionTable. For video sessions created via static configuration (e.g. via XML configuration file), the CCAP shall set mpegVideoSessionProvMethod to tableBased (1).

For video sessions created via dynamic signaling (e.g. via ERMI), the CCAP shall instantiate the appropriate row entries in the SCTE-HMS-MPEG-MIB's mpegProgramMappingTable, mpegVideoSessionTable, mpegVideoSessionPtrTable, and mpegInputTSOutputSessionTable. For video sessions created via dynamic signaling (e.g. via ERMI), the CCAP shall set mpegVideoSessionProvMethod to sessionBased (2).

The CCAP shall implement the mpegSessionsGroup table of SCTE-HMS-MPEG-MIB which is defined as optional in ANSI SCTE 154-5 [29].

The CCAP should support all optional MIB objects specified in the tables in Annex F: Detailed MIB Requirements.

For an example of identifying a replication QAM via the SCTE-HMS-MPEG-MIB, see clause I.1, Identifying Replicated QAMs.

7.1.1.4 CCAP MIB

The CCAP-MIB, included in the present document in Annex A, SNMP MIBs (Normative), defines the following:

- Objects which provide a link between an identifier of a CCAP interface used in the XML configuration file and its corresponding standard ifIndex MIB object from the ifTable and entPhysicalIndex MIB object from the Entity-MIB.
- Objects which can be used for video input program bitrate monitoring. Both the input program bitrate and input program requested bitrate can be accessed.

7.1.1.5 Specific MIB Object Implementation Requirements

7.1.1.5.1 SNMPv2-MIB System Group Requirements

The CCAP shall use the value of the Name attribute of the Ccap object when reporting sysName via the SNMPv2-MIB. The CCAP shall use the value of the Location attribute of the Ccap configuration object when reporting the sysLocation via the SNMPv2-MIB.

7.1.1.5.2 Interfaces Group MIB Requirements

The CCAP shall implement a row entry in the ifTable for each Downstream RF Port in the CCAP chassis. A Downstream RF Port is typically associated with a single F-connector or single MCX-75 connector on a DLC.

The CCAP shall 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 RF Port, the CCAP shall 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 RF Port, regardless of whether the QAM has been replicated.

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

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

The CCAP shall implement a row entry in the ifTable for each Upstream RF Port in the CCAP chassis. An Upstream RF Port is typically associated with a single F-connector or a single MCX-75 connector on a ULC.

The CCAP shall 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 RF Port, the CCAP shall automatically create one or more corresponding instances of an UpstreamLogicalChannel.

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

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

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

When an instance of DOCSIS MAC Domain is created, the CCAP shall create an ifTable entry with an ifType value of 127 (docsCableMaclayer).

For each loopback interface that is defined in the system, the CCAP shall represent that interface with an ifTable entry with an ifType value of 24, per RFC 2863 [11].

For each row entry created in the ifTable, the CCAP shall create a corresponding row entry in the ifXTable.

The CCAP 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 CCAP configuration UML object model.

7.1.1.5.2.1 CCAP ifStack Table

Shown in figure 7.1 is an example of how the ifStack table might look for downstream interfaces on the CCAP. The values used for the ifIndexes are for example purposes only. The ifStack table for the CCAP has been modified from previous versions of DOCSIS and CableLabs specifications. The rationale for this change is related to the multiservice nature of the CCAP and the desire to include the physical port in the ifStack. On the downstream side of the ifStack, the table remains consistent with the way Downstream Interfaces were modeled in the DOCSIS and Modular Headend Architectures, with the exception being the addition of the Downstream RF Port being placed at the bottom of the ifStack. The diagram in figure 7.1 shows both the VideoDownChannel objects and the DocsisDownChannel objects being sent over the same Downstream Radio Frequency (DS RF) Port.

On the upstream side, similar constructs have been used; however, the upstream model has inverted the relationship between Upstream Logical and Upstream Physical channels to more accurately reflect the nature of the relationships between burst receivers and the channels they are configured to receive. In the CCAP model, the lowest tier of the ifStack starts with the Upstream RF Port, then moves to the Upstream Physical Channel, and then progresses to the Upstream Logical Channels, and finally the DOCSIS MAC Domain.

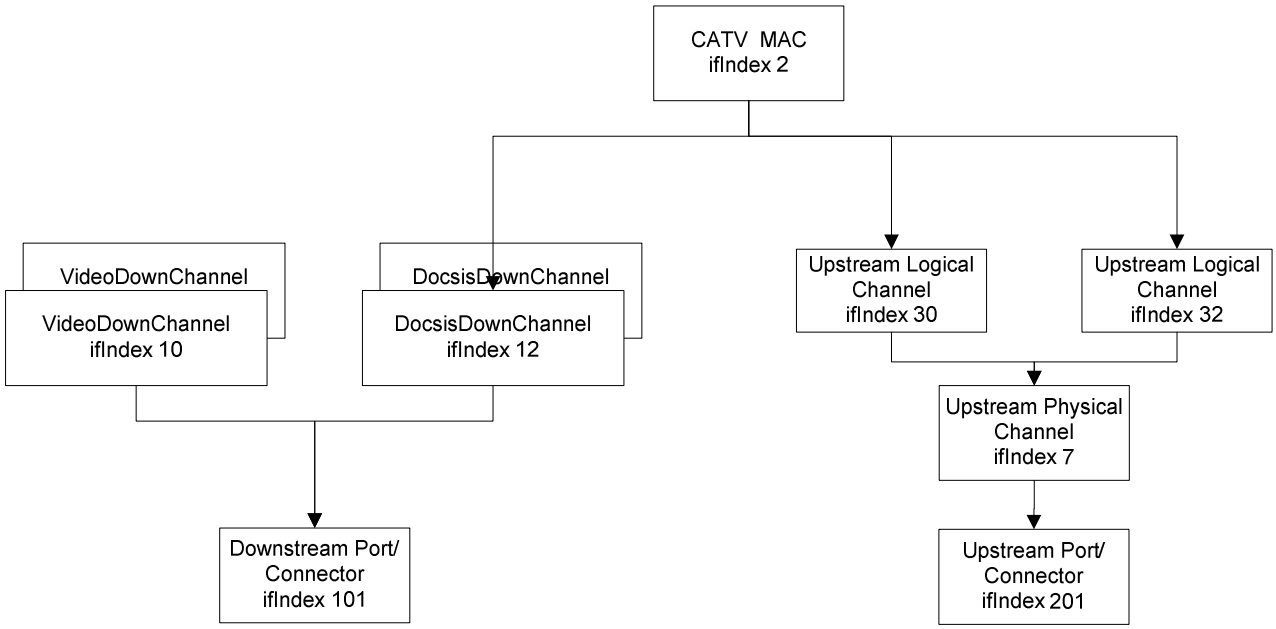


Figure 7.1: ifStack Table for CCAP RF Interfaces

Table 7.1: CCAP ifStack Table Representation

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

Table 7.2: IfTable/IfXTable Details for Ethernet Interfaces

MIB Objects	CCAP-Ethernet	DTI
IfTable		
ifIndex	(n)	(n)
ifDescr		
ifType	6	other(1)
ifMtu	1 500	256
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,	5
ifPhysAddress	MAC Address of this interface	Empty-String
ifAdminStatus For CCAP: When a managed system initializes, all interfaces start with ifAdminStatus in the up(1) state. As a result of either explicit management or configuration information saved via other non-SNMP method (i.e. CLI commands) retained by the managed system, ifAdminStatus is then changed to either the down(2) or testing(3) states (or remains in the up(1) state).	up(1), down(2), testing(3)	Up(1), Down(2), Testing(3)
ifOperStatus	Up(1), Down(2), Testing(3), Dormant(5), notPresent(6)	Up(1), Down(2), Testing(3), Dormant(5), notPresent(6)
ifLastChange		
ifXTable		
ifName		
ifLinkUpDownTrapEnable		
ifHighSpeed (mbits/sec)	100, 1 000, 10 000, 40 000, 100 000	5
ifPromiscuousMode	True(1), false(2)	True(1), false(2)
ifConnectorPresent		
ifAlias		
ifCounterDiscontinuityTime		

Table 7.3: IfTable/IfXTable for RF and DOCSIS® Interfaces

MIB Objects	CCAP-MAC	CCAP VideoDownChannel	CCAP DocsisDownChannel	CCAP-Upstream Physical Channel	CCAP-Upstream Logical Channel	CCAP DsRfPort	CCAP UsRfPort
IfTable							
ifIndex	(n)	(n)	(n)	(n)	(n)	(n)	(n)
ifDescr							
ifType	127	214*	128	129	205	257	256
ifMtu For RF Upstream/Downstream; the value includes the length of the MAC header.	1 522	188	1 764	1 764	1 764	0	0
ifSpeed For CCAP VideoDownChannels and DocsisDownChannels; This is the symbol rate multiplied by the number of bits per symbol. For RF Upstream; this is the raw band-width in bits per second of this interface, regarding 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.	0	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
ifPhysAddress	MAC Address of this interface	Empty-String	Empty-String	Empty-String	Empty-String	Empty-String	Empty-String

MIB Objects	CCAP-MAC	CCAP VideoDownChannel	CCAP DocsisDownChannel	CCAP-Upstream Physical Channel	CCAP-Upstream Logical Channel	CCAP DsRfPort	CCAP UsRfPort
IfTable							
ifAdminStatus: For CCAP: When a managed system initializes, all interface start with ifAdminStatus in the down(2) state. As a result of either explicit management or configuration information saved via other non SNMP method (i.e. CLI commands) retained by the managed system, ifAdminStatus is then changed to either the up(1) or testing(3) states (or remains in the down(2) state).	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	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							
ifHighSpeed For CCAP Video DownChannel and DocsisDownChannel; this is the symbol rate multiplied with the number of bits per symbol. For RF Upstream; This is the raw bandwidth in bits per second of this interface, regarding 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.	0	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

MIB Objects	CCAP-MAC	CCAP VideoDownChannel	CCAP DocsisDownChannel	CCAP-Upstream Physical Channel	CCAP-Upstream Logical Channel	CCAP DsRfPort	CCAP UsRfPort
IfTable							
ifPromiscuousMode	True(1), False(2)		False(2)	True(1), False(2)	True(1)	False(2)	False(2)
ifConnectorPresent							
ifAlias							
ifCounterDiscontinuityTime							
NOTE: Also considered 226-QAM, but selected MPEG transport because the interface represents the logical content rather than the physical transmission.							

Table 7.4: CCAP ifCounters Information

MIB Counter Objects	Access	CCAP-MAC	CCAP-VideoDown Channel	CCAP-DocsisDown Channel	CCAP-Upstream Physical Channel	CCAP-Upstream Logical Channel	CCAP-DsRfPort	CCAP-UsRfPort
ifTable								
ifInOctets 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. For MAC; The total number of data octets (data in transit, data targeted to the managed device) received on this interface from the RF interface and before application of protocol filters.	RO	Mandatory	Mandatory	NA	Mandatory	Mandatory	NA	NA
ifInUcastPkts For RF Upstream/Downstream; Reports zero if implemented. For MAC; the total number of Unicast data packets (data in transit, data targeted to the managed device) received on this interface from the RF interface before application of protocol filters.	RO	Mandatory	NA	NA	Optional	Optional	NA	NA
ifInDiscards	RO	Mandatory	Mandatory	NA	Optional	Optional	NA	NA
ifInErrors	RO	Mandatory	Mandatory	NA	Optional	Optional	NA	NA
ifInUnknownProtos	RO	Mandatory	NA	NA	Optional	Optional	NA	NA

MIB Counter Objects	Access	CCAP-MAC	CCAP-VideoDown Channel	CCAP-DocsisDown Channel	CCAP-Upstream Physical Channel	CCAP-Upstream Logical Channel	CCAP-DsRfPort	CCAP-UsRfPort
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. For MAC; The total number of data octets (data in transit, data generated by the managed device) transmitted on this interface to the RF interface after application of protocol filters.	RO	Mandatory	Mandatory	M	NA	NA	NA	NA
ifOutUcastPkts For RF Upstream/Downstream; Reports zero if implemented. For MAC; The total number of Unicast data packets (data in transit, data generated by the managed device) transmitted on this interface to the RF interface after application of protocol filters.	RO	Mandatory	NA	O	NA	NA	NA	NA
ifOutDiscards	RO	Mandatory	NA	O	NA	NA	NA	NA
ifOutErrors	RO	Mandatory	NA	O	NA	NA	NA	NA
ifXTable			NA				NA	NA
ifInMulticastPkts For RF Upstream/Downstream; Reports zero if implemented. For MAC; the total number of Multicast data packets (data in transit, data targeted to the managed device) received on this interface from the RF interface before application of protocol filters.	RO	Mandatory	NA	NA	Optional	Optional	NA	NA
ifInBroadcastPkts For RF Upstream/Downstream; Reports zero if implemented. For MAC; The total number of Broadcast data packets (data in transit, data targeted to the managed device) received on this interface from the RF interface before application of protocol filters.	RO	Mandatory	NA	NA	Optional	Optional	NA	NA

MIB Counter Objects	Access	CCAP-MAC	CCAP-VideoDown Channel	CCAP-DocsisDown Channel	CCAP-Upstream Physical Channel	CCAP-Upstream Logical Channel	CCAP-DsRfPort	CCAP-UsRfPort
ifOutMulticastPkts For RF Upstream/Downstream; Reports zero if implemented. For MAC; The total number of Multicast data packets (data in transit, data generated by the managed device) transmitted on this interface to the RF interface after application of protocol filters.	RO	Mandatory	NA	O	NA	NA	NA	NA
ifOutBroadcastPkts For RF Upstream/Downstream; Reports zero if implemented. For MAC; The total number of Broadcast data packets (data in transit, data generated by the managed device) transmitted on this interface to the RF interface after application of protocol filters.	RO	Mandatory	NA	O	NA	NA	NA	NA
IfHCInOctets 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. For MAC; The total number of data octets (data in transit, data targeted to the managed device) received on this interface from the RF interface and before application of protocol filters.	RO	Mandatory	Mandatory	NA	Mandatory	Mandatory	NA	NA
ifHCInUcastPkts For RF Upstream/Downstream; Reports zero if implemented. For MAC; the total number of Unicast data packets (data in transit, data targeted to the managed device) received on this interface from the RF interface before application of protocol filters.	RO	Optional	NA	NA	Optional	Optional	NA	NA

MIB Counter Objects	Access	CCAP-MAC	CCAP-VideoDown Channel	CCAP-DocsisDown Channel	CCAP-Upstream Physical Channel	CCAP-Upstream Logical Channel	CCAP-DsRfPort	CCAP-UsRfPort
ifHCInMulticastPkts For RF Upstream/Downstream; Reports zero if implemented. For MAC; the total number of Multicast data packets (data in transit, data targeted to the managed device) received on this interface from the RF interface before application of protocol filters.	RO	Optional	NA	NA	Optional	Optional	NA	NA
ifHCInBroadcastPkts For RF Upstream/Downstream; Reports zero if implemented. For MAC; The total number of Broadcast data packets (data in transit, data targeted to the managed device) received on this interface from the RF interface before application of protocol filters.	RO	Optional	NA	NA	Optional	Optional	NA	NA
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. For MAC; The total number of data octets (data in transit, data generated by the managed device) transmitted on this interface to the RF interface after application of protocol filters.	RO	Mandatory	Mandatory	M	NA	NA	NA	NA
ifHCOutUcastPkts For RF Upstream/Downstream; Reports zero if implemented. For MAC; The total number of Unicast data packets (data in transit, data generated by the managed device) transmitted on this interface to the RF interface after application of protocol filters.	RO	Optional	NA	O	NA	NA	NA	NA

MIB Counter Objects	Access	CCAP-MAC	CCAP-VideoDown Channel	CCAP-DocsisDown Channel	CCAP-Upstream Physical Channel	CCAP-Upstream Logical Channel	CCAP-DsRfPort	CCAP-UsRfPort
ifHCOutMulticastPkts For RF Upstream/Downstream; Reports zero if implemented. For MAC; The total number of Multicast data packets (data in transit, data generated by the managed device) transmitted on this interface to the RF interface after application of protocol filters.	RO	Optional	NA	O	NA	NA	NA	NA
ifHCOutBroadcastPkts For RF Upstream/Downstream; Reports zero if implemented. For MAC; The total number of Broadcast data packets (data in transit, data generated by the managed device) transmitted on this interface to the RF interface after application of protocol filters.	RW	Optional	NA	O	NA	NA	NA	NA

7.1.1.5.3 Entity-MIB Requirements

The CCAP shall implement a row entry in the entPhysicalTable for each the system chassis and Field Replaceable Unit (FRU) installed in the CCAP chassis.

The CCAP shall 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, fabric cards, DTI cards, SREs, DLCs, ULCs, combined Upstream & Downstream line cards, Ethernet cards, and PON line cards.

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

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

The CCAP 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 CCAP shall create a corresponding row entry in the entAliasMappingTable.

The CCAP shall implement a row entry in the entAliasMappingTable for each UsRfPort and each DsRfPort in the chassis.

The CCAP 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.

7.2 Performance Management UML Object Models

The Performance Management UML model has been divided into the following categories:

- State Data: These object are used to gather state information from the CCAP.
- Statistical Data: These objects are used to gather statistical information from the CCAP.

Those models are shown in the following sections.

7.2.1 State Data Objects

7.2.1.1 DOCS-IF3-MIB: CMTS Bonding

The objects in the DOCS-IF3-MIB: CMTS Bonding are taken from the DOCS-IF3-MIB specified in Annex Q of [7] and used without modification for the CCAP.

Reference: [7], DOCS-IF3-MIB.

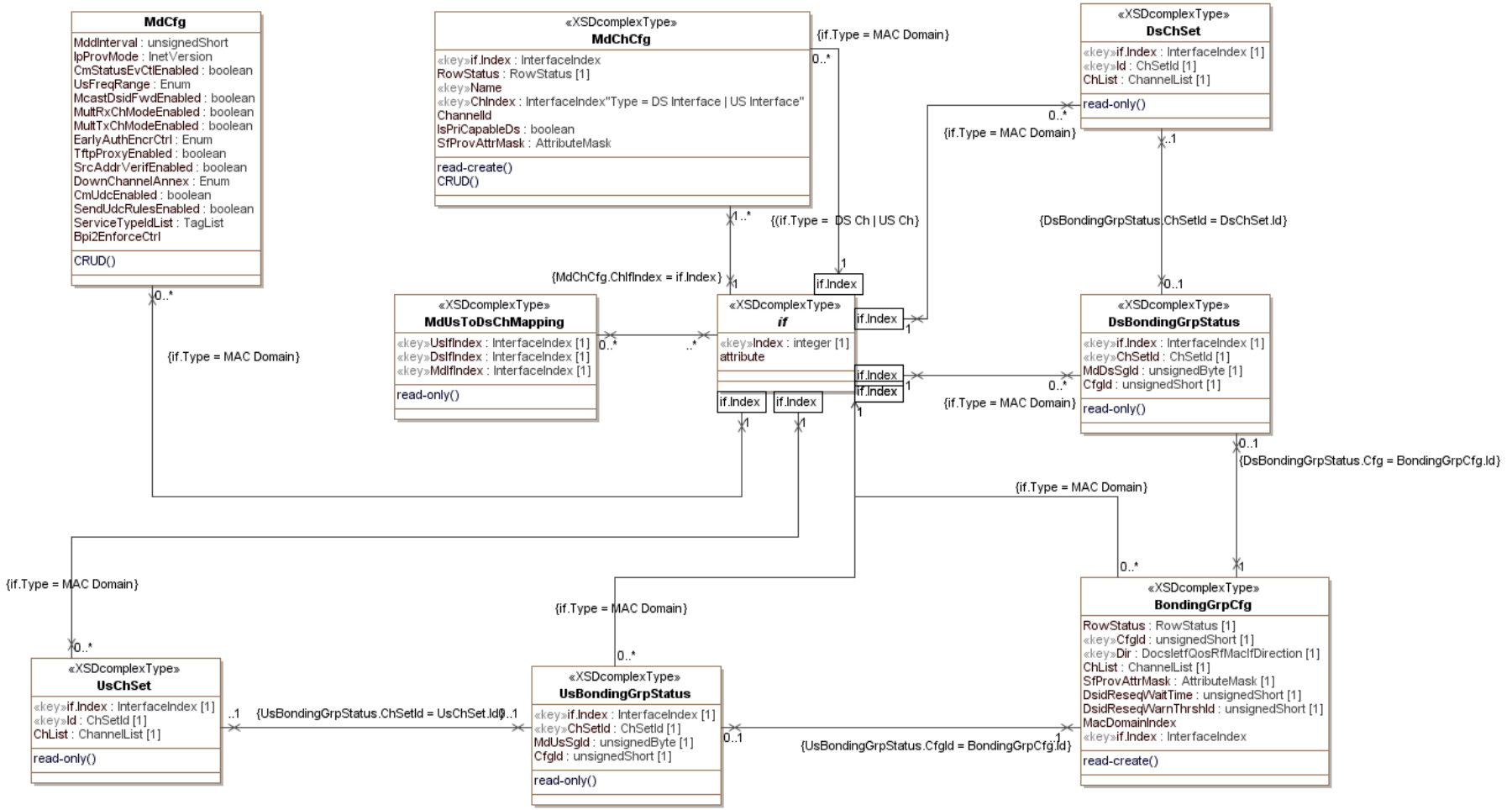


Figure 7.2: DOCS-IF3-MIB: CMTS Bonding Performance Management Objects

7.2.1.2 DOCS-IF3-MIB: RxCh Objects

In the following diagram, the majority of the objects is taken from the DOCS-IF3-MIB, specified in Annex Q of [7], and used without modification. The RccCfg object is taken from the CCAP Configuration UML model, described in clause 6.5.7.6.14.

Reference: [7], DOCS-IF3-MIB

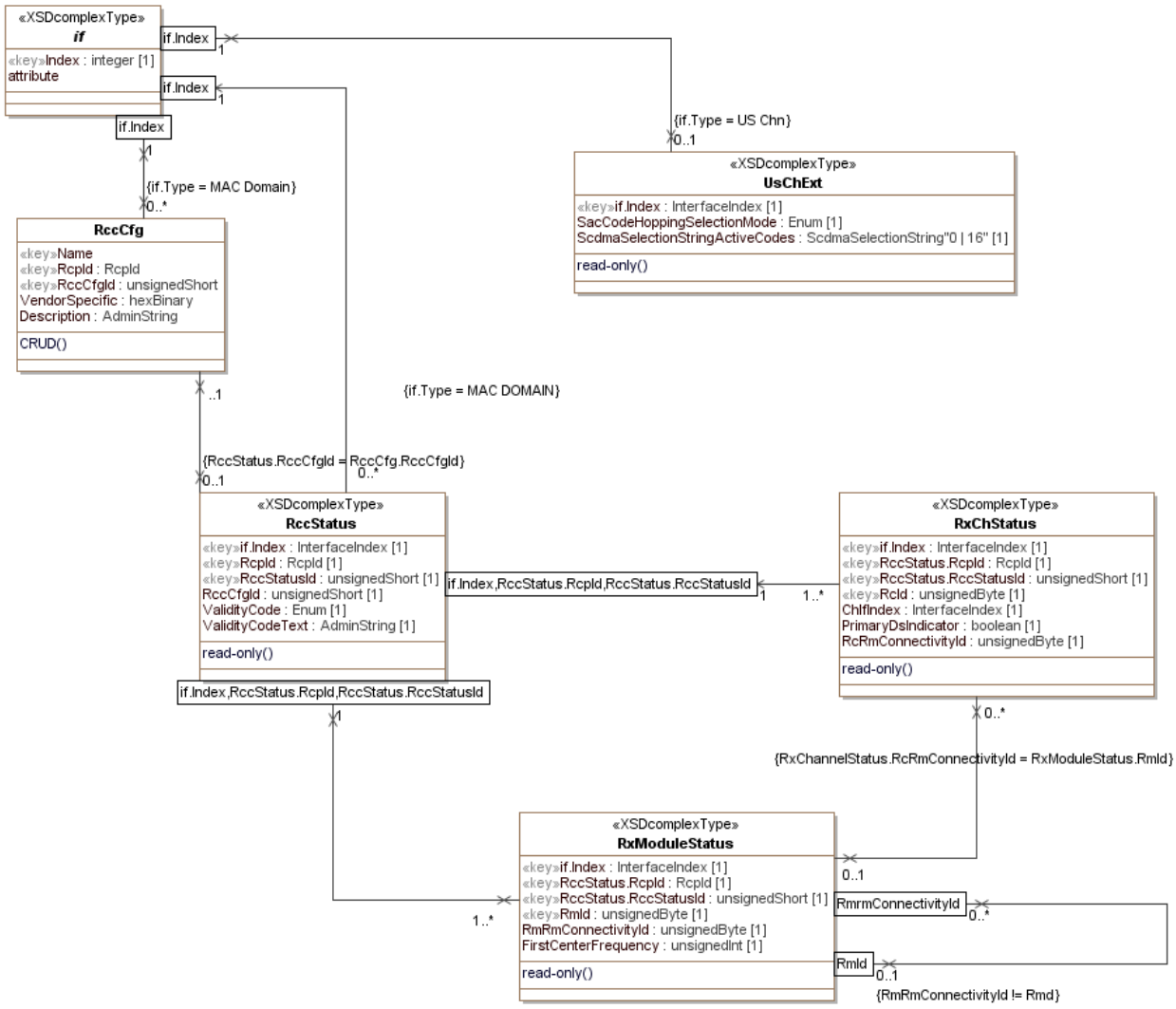


Figure 7.3: DOCS-IF3-MIB: RxCh Performance Management Objects

7.2.1.3 DOCS-L2VPN-MIB State Objects

The objects in the DOCS-L2VPN-MIB: State Objects are taken from the DOCS-L2VPN-MIB specified in Annex A of ES 203 085 [3] and are used without modification for the CCAP.

Reference: ES 203 085 [3], DOCS-L2VPN-MIB.

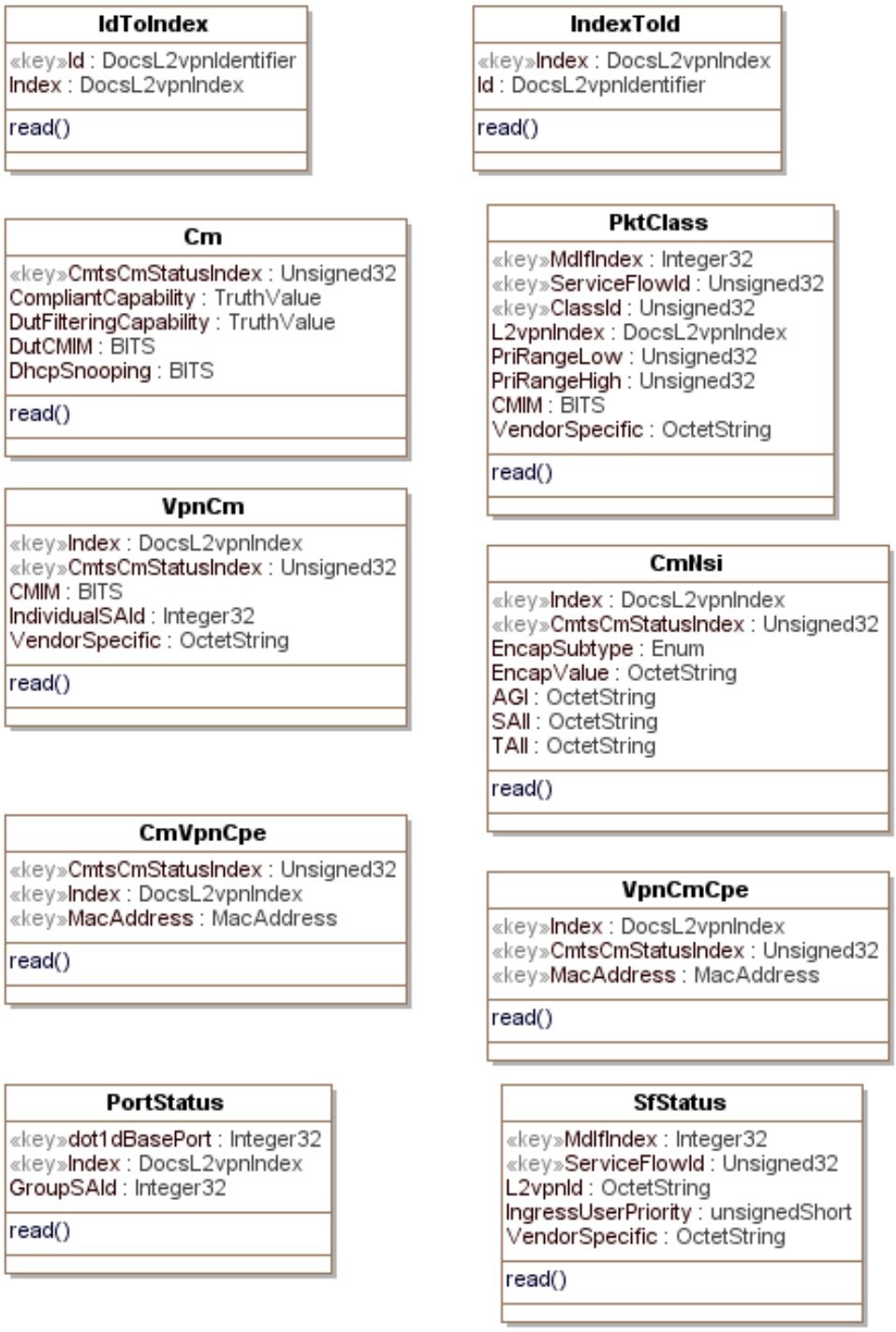


Figure 7.4: DOCS-L2VPN-MIB: State Objects

7.2.1.4 DOCS-LOADBAL3-MIB

The objects in the DOCS-LOADBAL3-MIB are taken from the DOCS-LOADBAL3-MIB specified Annex Q of [7] and used without modification for the CCAP.

The following attributes of the CmtsCmParams object are writeable:

- ProvGrpId
- ProvServiceTypeId
- PolicyId

- Priority

Reference: [7], DOCS-LOADBAL3-MIB.

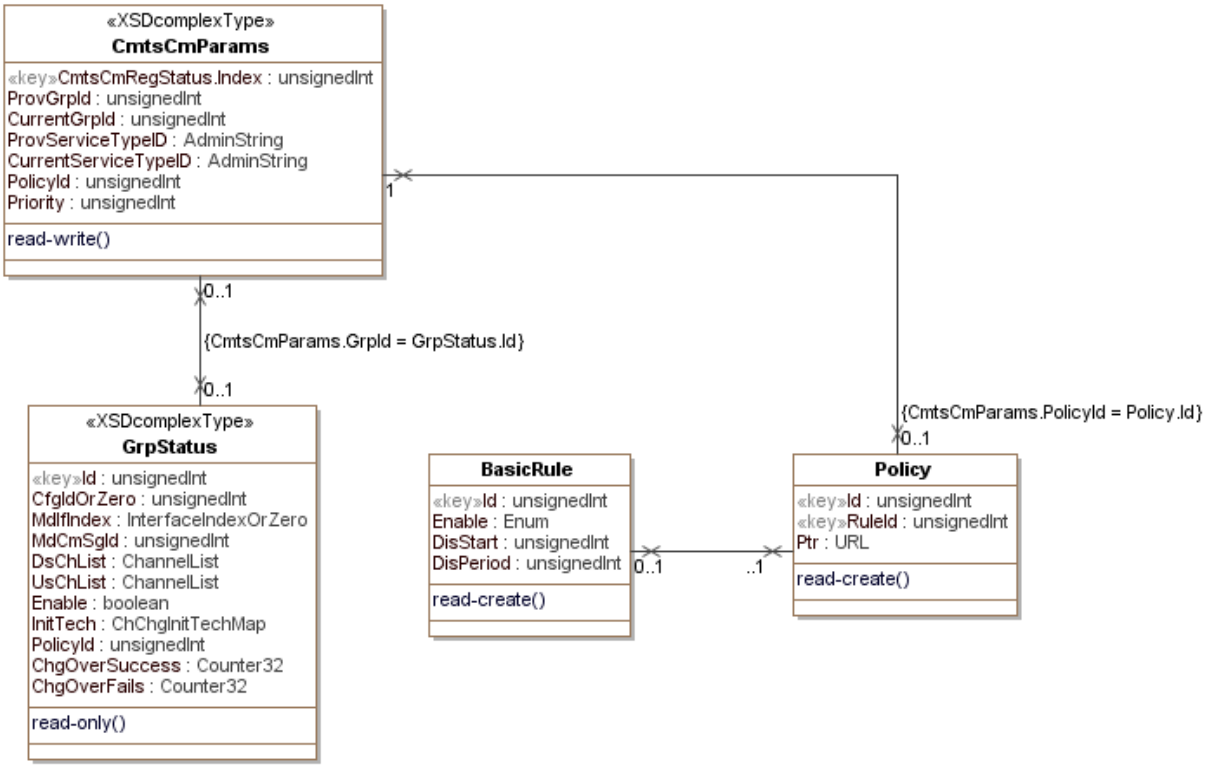


Figure 7.5: DOCS-IF3-MIB: RxCh Performance Management Objects

7.2.1.5 DOCS-MCAST-AUTH-MIB

The objects in the DOCS-MCAST-AUTH-MIB are taken from the DOCS-MCAST-AUTH-MIB specified in Annex Q of [7] and used without modification for the CCAP.

Reference: [7], DOCS-MCAST-AUTH-MIB.

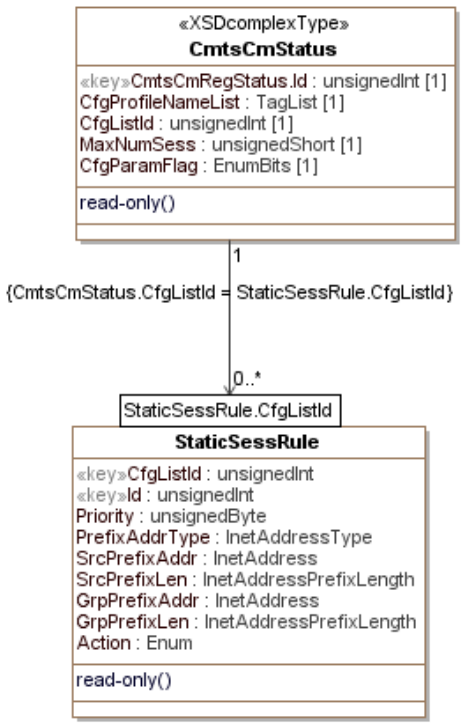


Figure 7.6: DOCS-MCAST-AUTH-MIB Performance Management Objects

7.2.1.6 DOCS-QOS3-MIB: State Objects

The objects in the DOCS-QOS3-MIB: State Objects are taken from the DOCS-QOS3-MIB specified in Annex Q of [7] and used without modification for the CCAP.

Reference: [7], DOCS-QOS3-MIB.

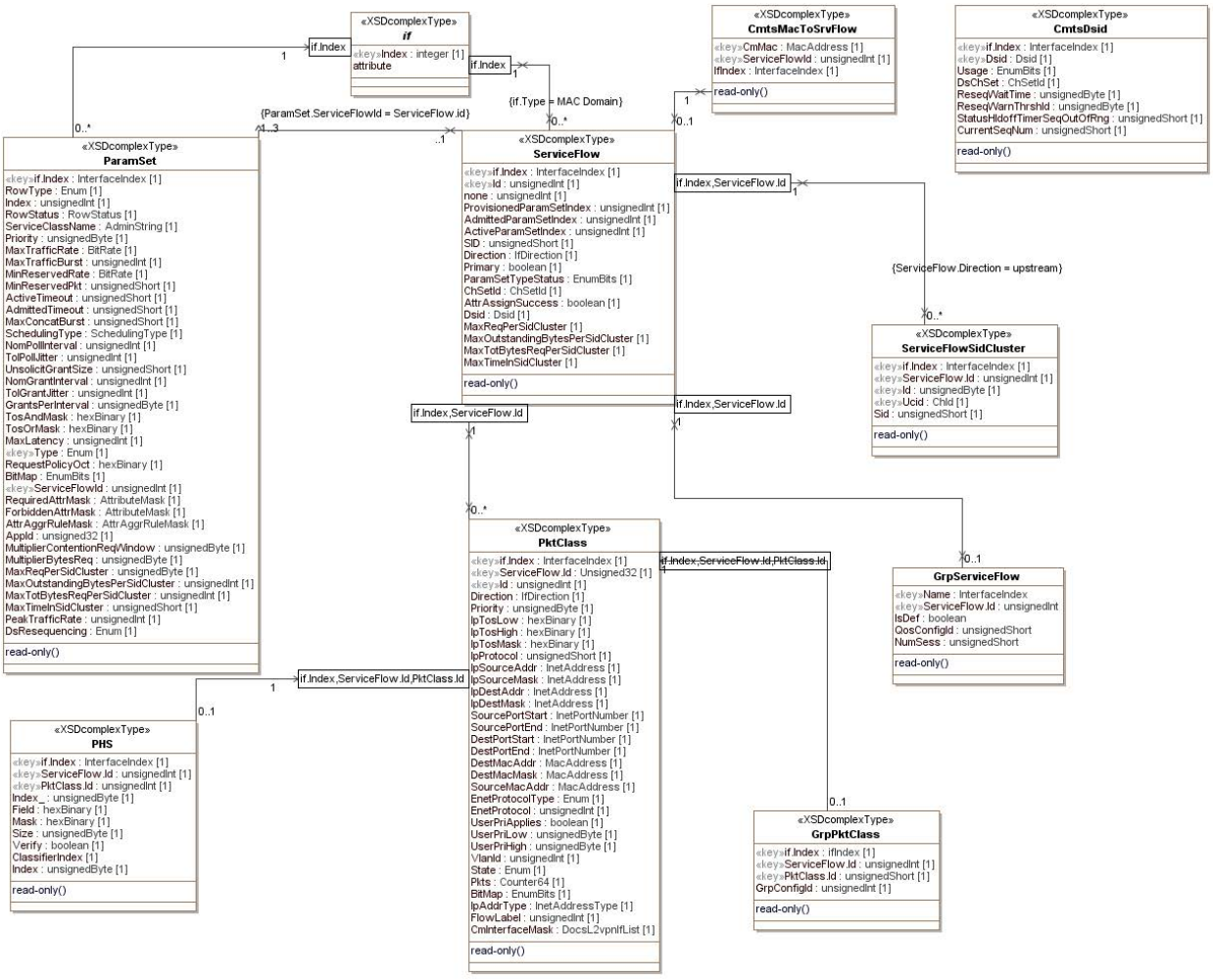


Figure 7.7: DOCS-QOS3-MIB: State Objects Performance Management Objects

7.2.1.7 DOCS-SEC-MIB

The objects in the DOCS-SEC-MIB are taken from the DOCS-SEC-MIB specified in Annex Q of [7]; the DocsSecCmtsCertRevocationListStatus object only includes the read-only attributes. Otherwise, these objects are used without modification for the CCAP.

Reference: [7], DOCS-SEC-MIB.

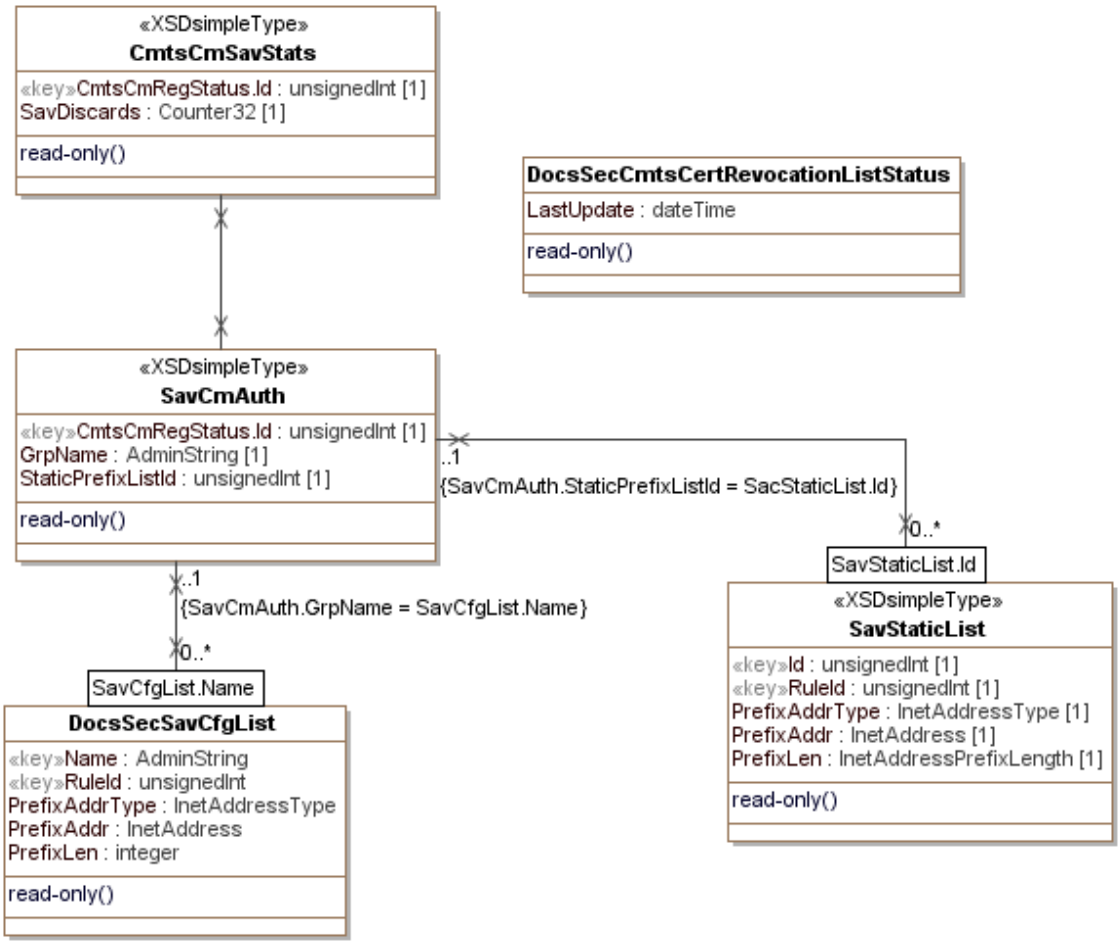


Figure 7.8: DOCS-SEC-MIB Performance Management Objects

7.2.1.8 DOCS-SUBMGT3-MIB

The following objects in the DOCS-SUBMGT3-MIB are taken from the DOCS-IF3-MIB specified in Annex Q of [7] and used without modification for the CCAP:

- CpeIp
- CpeCtrl
- CmtsCmRegStatus

The Grp object in the DOCS-SUBMGT3-MIB is taken from the DOCS-SUBMGT3-MIB specified in Annex Q of [7] and used without modification for the CCAP.

The DocsSubMgmt3FilterGrp object is taken from the CCAP Configuration UML model, described in clause 6.5.7.3.5.

Reference: [7], DOCS-SUBMGT3-MIB.

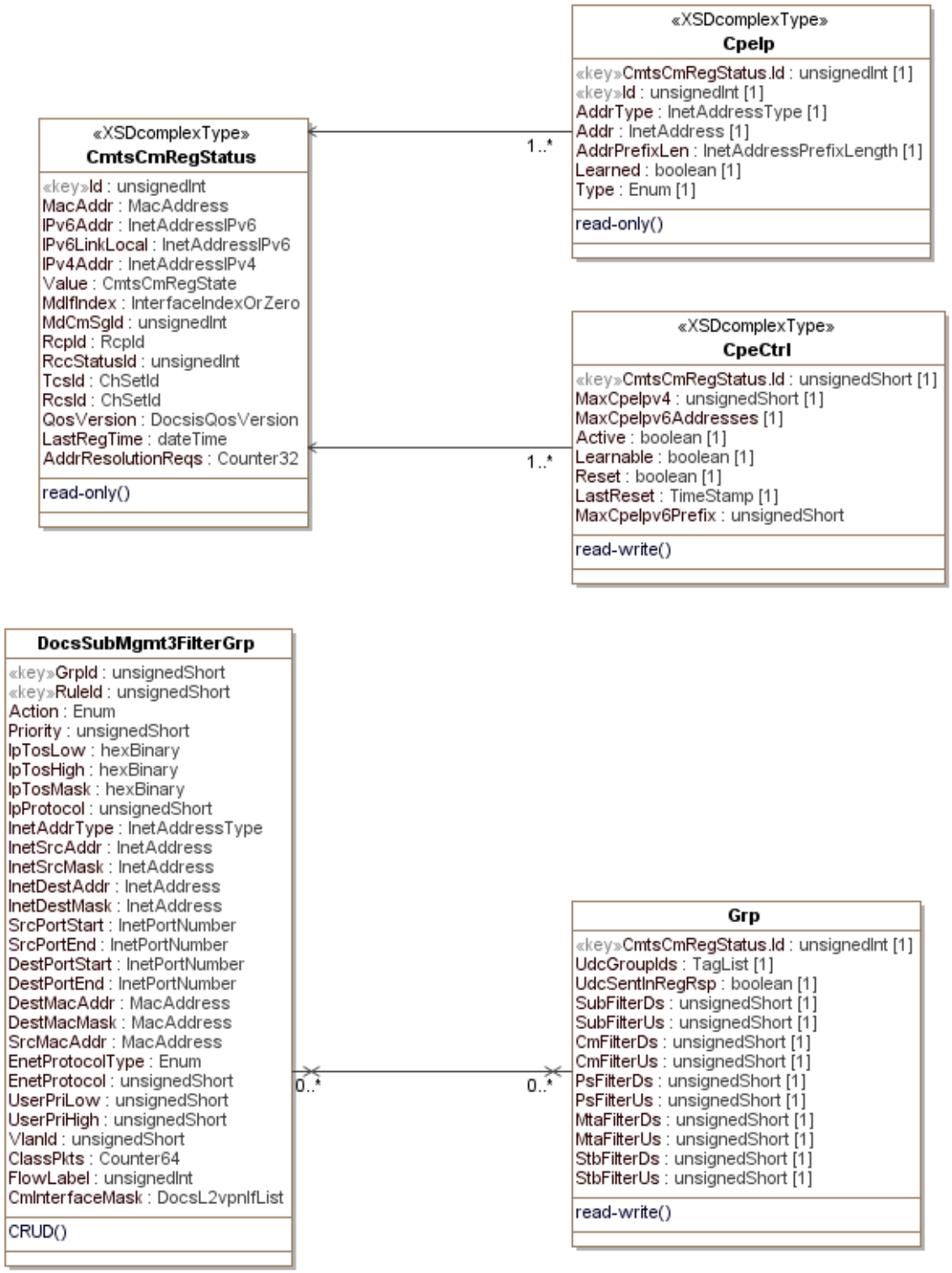


Figure 7.9: DOCS-MCAST-MIB Performance Management Objects

7.2.1.9 CCAP Topology Objects

The following CCAP topology objects are taken from the DOCS-IF3-MIB specified in Annex Q of [7] and used without modification for the CCAP:

- MdNodeStatus
- MdDsSgStatus

- MdUsSgStatus

The RfPortFnCfg object is taken from the CLAB-TOPO-MIB specified in Annex Q of [7] and used without modification for the CCAP.

The FiberNodeCfg object is taken from the CCAP Configuration UML model; it is defined in clause 6.5.5.18.

Reference: [7], DOCS-IF3-MIB, CLAB-TOPO-MIB.

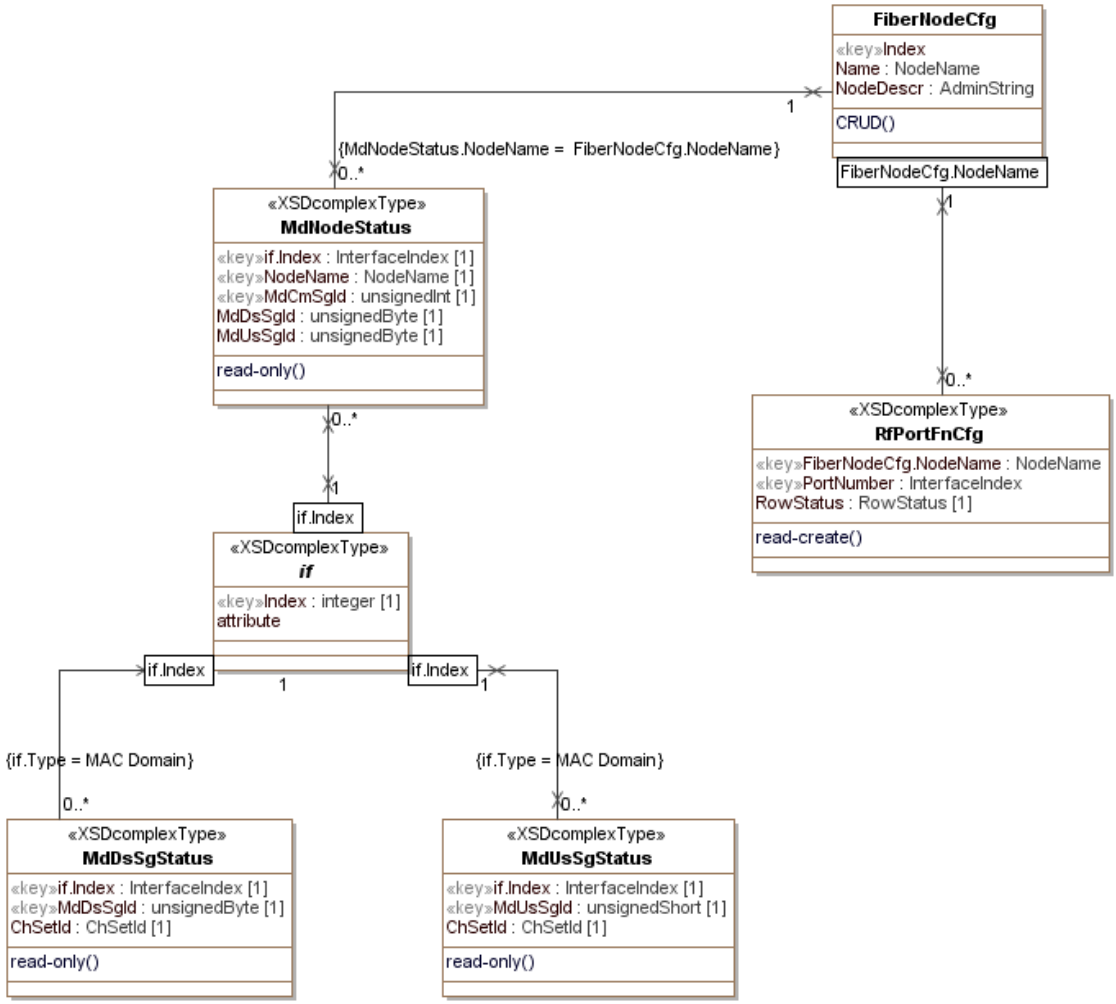


Figure 7.10: CCAP Topology Performance Management Objects

7.2.1.10 CCAP-MIB

The CCAP-MIB defines the following:

- Objects that provide a link between an identifier of a CCAP interface used in the XML configuration file and its corresponding standard ifIndex MIB object from the ifTable and entPhysicalIndex MIB object from the ENTITY-MIB.
- Objects that can be used for video input program bitrate monitoring. Both the input program bitrate and input program requested bitrate can be accessed.
- Objects that can be used to determine the status of the ECMD and ECMG.

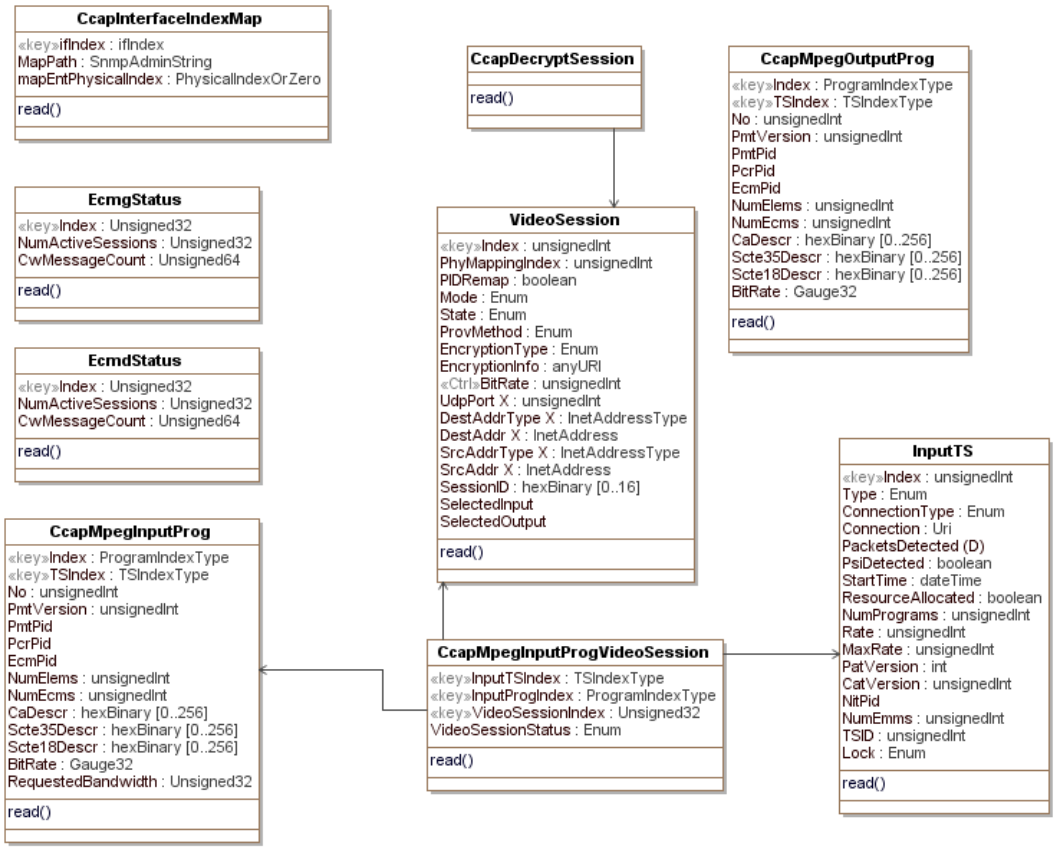


Figure 7.11: CCAP-MIB Performance Management Objects

The objects that make up the CCAP-MIB are described in the following clauses.

7.2.1.10.1 CcapInterfaceIndexMap

This object reports the corresponding device path for the Interface index defined by an object instance.

Table 7.5: CcapInterfaceIndexMap Object Attributes

Attribute Name	Type	Access	Type Constraints	Units	Default Value
ifIndex	ifIndex	key			
MapPath	SnmpAdminString	read-only			
mapEntPhysicalIndex	PhysicalIndexOrZero	read-only			

7.2.1.10.1.1 CcapInterfaceIndexMap Attributes

7.2.1.10.1.1.1 ifIndex

The index corresponds to the Interface MIB index for interfaces of IANA interface types:

- MAC Interface: docsCableMaclayer - 127
- Downstream Channel: docsCableDownstream - 128
- Upstream Interface: docsCableUpstream - 129
- Logical Upstream Channel: docsCableUpstreamChannel - 205
- Upstream RF Port: docsCableUpstreamRfPort - 256
- Downstream RF Port: cableDownstreamRfPort - 257

7.2.1.10.1.1.2 MapPath

This attribute indicates the CCAP node XPath expression that identifies the resource associated with the interface index. For example, the path value of the resource associated with an upstream logical channel with index = 5, in upstream physical channel index = 7, in an Upstream RF port number = 15, from an US RF Line Card, in slot number = 3, chassis id = 1 is represented as:

```
/ccap/chassis[id="1"]
/slot[number="3"]
/rf-line-card
/us-rf-port[number="15"]
/upstream-physical-channel[index="7"]
/upstream-logical-channel[index="5"]
```

NOTE: Line breaks in this example were added for clarity.

7.2.1.10.1.1.3 mapEntPhysicalIndex

This attribute corresponds to the entPhysical Index associated with the resource. The value is zero (0) if undefined.

7.2.1.10.2 EcmgStatus

This object allows for the monitoring of the interface to an Entitlement Control Message Generator (ECMG).

Table 7.6: EcmgStatus Object Attributes

Attribute Name	Type	Access	Type Constraints	Units	Default Value
Index	UnsignedInt	Key			
NumActiveSessions	UnsignedInt	read-only			
CwMessageCount	UnsignedLong	read-only			

7.2.1.10.2.1 EcmgStatus Object Attributes

7.2.1.10.2.1.1 Index

This is an index for an instance of this object. It is a pointer to a defined Ecmg object.

7.2.1.10.2.1.2 NumActiveSessions

The current number of encryption sessions managed by the ECMG.

7.2.1.10.2.1.3 CwMessageCount

A running 64-bit counter that increments by one, every time the Encryptor receives one CW message from the ECMG. The counter is reset at boot time.

7.2.1.10.3 EcmdStatus

This object allows for the monitoring of the interface to an Entitlement Control Message Decoder (ECMD).

Table 7.7: EcmdStatus Object Attributes

Attribute Name	Type	Access	Type Constraints	Units	Default Value
Index	UnsignedInt	Key			
NumActiveSessions	UnsignedInt	read-only			
CwMessageCount	UnsignedLong	read-only			

7.2.1.10.3.1 EcmdStatus Object Attributes

7.2.1.10.3.1.1 EcmdIndex

This is an index for an instance of this object. It is a pointer to a defined Ecmd object.

7.2.1.10.3.1.2 NumActiveSessions

The current number of decryption sessions managed by the ECMD.

7.2.1.10.3.1.3 CwMessageCount

A running 64-bit counter that increments by one, every time the Decryptor receives one CW message from the ECMD. The counter is reset at boot time.

7.2.1.10.4 CcapMpegInputProg

This object augments the mpegInputProgTable of the SCTE-HMS-MPEG-MIB with two additional attributes:

- BitRate
- RequestedBandwidth

No further modifications have been made to this table.

Reference: ANSI SCTE 154-4 [29].

Table 7.8: CcapMpegInputProg Object Attributes

Attribute Name	Type	Access	Type Constraints	Units	Default Value
BitRate	Gauge32	read-only		BPS	
RequestedBandwidth	UnsignedInt	read-only			

7.2.1.10.4.1 CcapMpegInputProg Object Attributes

7.2.1.10.4.1.1 BitRate

Indicates the measured MPEG input program bitrate in bits per second.

7.2.1.10.4.1.2 RequestedBandwidth

Requested bandwidth for this MPEG input program. This value is used to validate the total QAM bandwidth before allowing the creation of a new session. It is also used to validate the input program bandwidth overflow situation during the transmission. In the case of special stream without PCR, it is used to limit the output bandwidth of that special program.

A zero (0) value is returned if no bandwidth validation is done on this program.

7.2.1.10.5 CcapMpegOutputProg

This object augments the mpegOutputProgTable of the SCTE-HMS-MPEG-MIB with the addition of a BitRate attribute.

No further modifications have been made to this table.

Reference: ANSI SCTE 154-4 [29].

Table 7.9: CcapMpegOutputProg Object Attributes

Attribute Name	Type	Access	Type Constraints	Units	Default Value
BitRate	Gauge32	read-only		BPS	

7.2.1.10.5.1 CcapMpegOutputProg Object Attributes

7.2.1.10.5.1.1 BitRate

Indicates the measured MPEG output program bitrate in bits per second.

7.2.1.10.6 VideoSession

The VideoSession object is taken from the SCTE-HMS-MPEG-MIB specified in ANSI SCTE 154-4 [29] and used without modification for the CCAP.

7.2.1.10.7 CcapDecryptSession

The CcapDecryptSession extends the existing VideoSession object from the SCTE-HMS-MPEG-MIB specified in ANSI SCTE 154-4 [29] and used without modification for the CCAP. This table is only populated with video sessions that require CCAP decryption.

Reference: ANSI SCTE 154-4 [29].

7.2.1.10.8 CcapMpegInputProgVideoSession

This object reports the list of video sessions that the MPEG input program are feeding.

Table 7.10: CcapMpegInputProgVideoSession Object Attributes

Attribute Name	Type	Access	Type Constraints	Units	Default Value
InputTSIndex	TSIndexType	key			
InputProgIndex	ProgramIndexType	key			
VideoSessionIndex	UnsignedInt	key			
VideoSessionStatus	Enum	read-only	active(1), closed(2)		

Table 7.11: CcapMpegInputProgVideoSession Object Associations

Associated Object Name	Type	Near-end Multiplicity	Far-end Multiplicity	Label
CcapMpegInputProg	Directed association to CcapMpegInputProg			
VideoSession	Directed association to VideoSession			
InputTS	Directed association to InputTS			

7.2.1.10.8.1 CcapMpegInputProgVideoSession Object Attributes

7.2.1.10.8.1.1 InputTSIndex

The index of the input TS.

7.2.1.10.8.1.2 InputProgIndex

The index of the input program.

7.2.1.10.8.1.3 VideoSessionIndex

The index of the video session.

7.2.1.10.8.1.4 VideoSessionStatus

The status of the video session.

7.2.1.10.9 InputTS

The InputTS object is taken from the SCTE-HMS-MPEG-MIB specified in ANSI SCTE 154-4 [29] and used without modification for the CCAP.

Reference: ANSI SCTE 154-4 [29].

7.2.1.11 SCTE-HMS-MPEG-MIB: State Objects

The objects in the SCTE-HMS-MPEG-MIB: State Objects are taken from ANSI SCTE 154-4 [29] and used with the following modifications for the CCAP.

The CcapMpegInputProg object replaces the MpegInputProg object from the SCTE-HMS-MPEG-MIB. It is defined in clause 7.2.1.10.4.

The CcapMpegOutputProg object replaces the MpegOutputProg object from the SCTE-HMS-MPEG-MIB. It is defined in clause 7.2.1.10.5.

Reference: ANSI SCTE 154-4 [29].

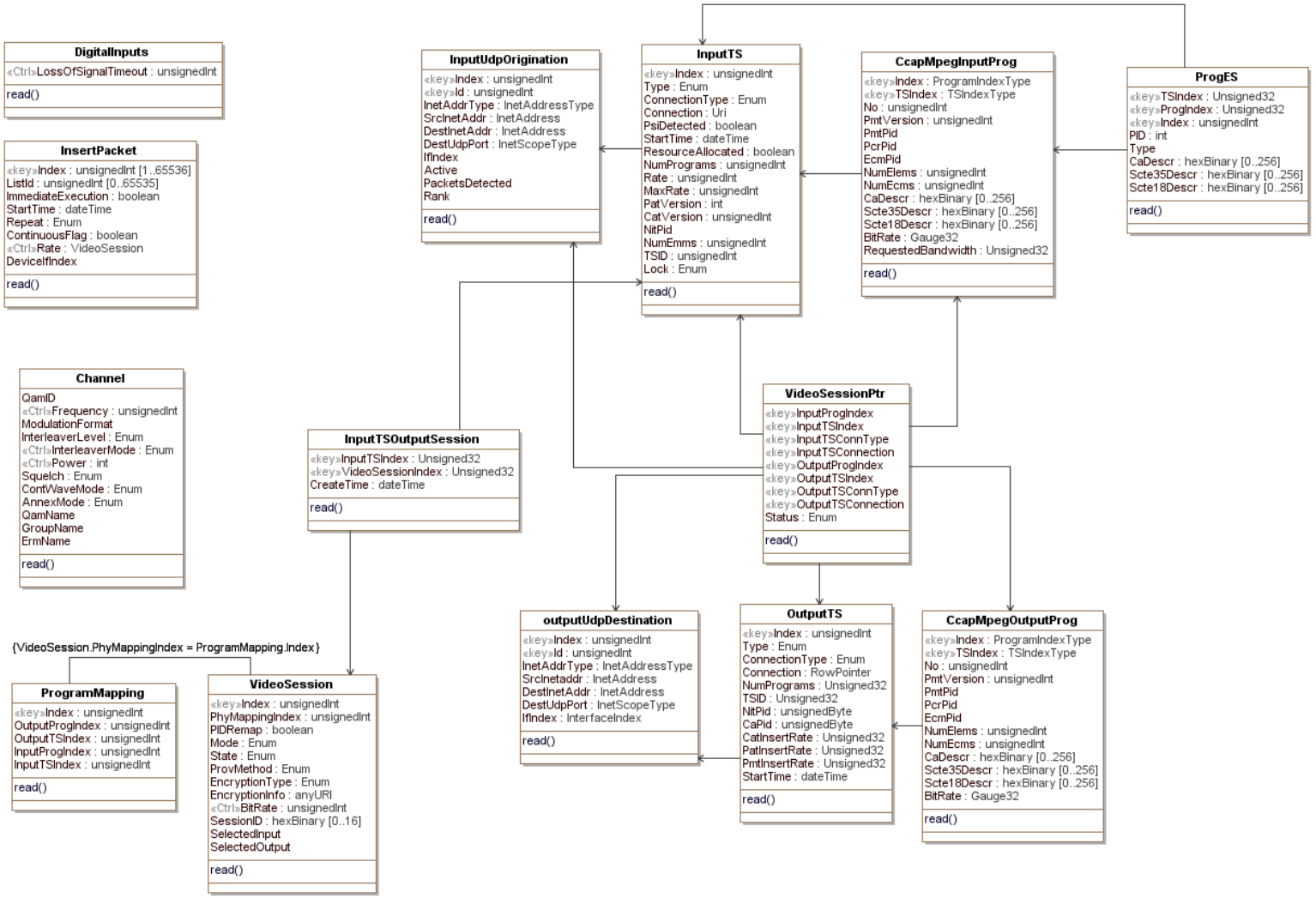


Figure 7.12: SCTE-HMS-MPEG-MIB: State Objects Performance Management Objects

7.2.1.12 DOCS-DRF-MIB

The objects in the DOCS-DRF-MIB: State Objects are taken from the DOCS-DRF-MIB specified in Annex A of [4] and used without modification for the CCAP.

Reference: [4], DOCS-DRF-MIB.

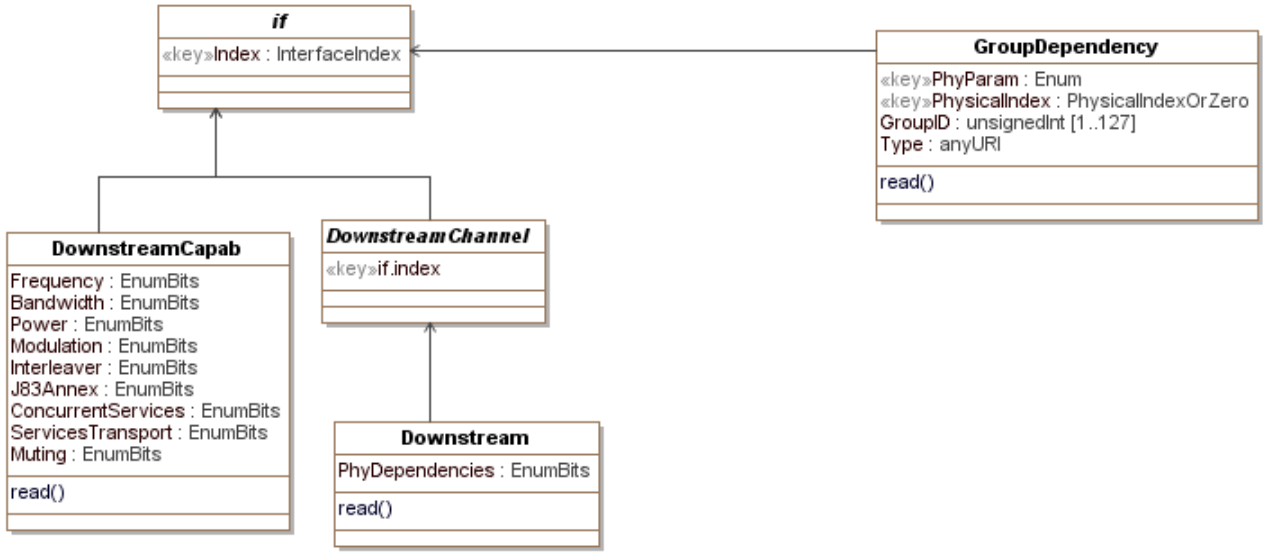


Figure 7.13: DOCS-DRF-MIB Performance Management Objects

7.2.2 Statistical Data Objects

7.2.2.1 DOCS-IF-MIB

The objects in the DOCS-IF-MIB are taken from RFC 4546 [20] and used without modification for the CCAP.

Reference: RFC 4546 [20].

Figure 7.14: DOCS-IF-MIB Performance Management Objects

7.2.2.2 DOCS-IF3-MIB

The objects in the DOCS-IF3-MIB are taken from the DOCS-IF3-MIB specified in Annex Q of [7] and used without modification for the CCAP.

Reference: [7], DOCS-IF3-MIB.

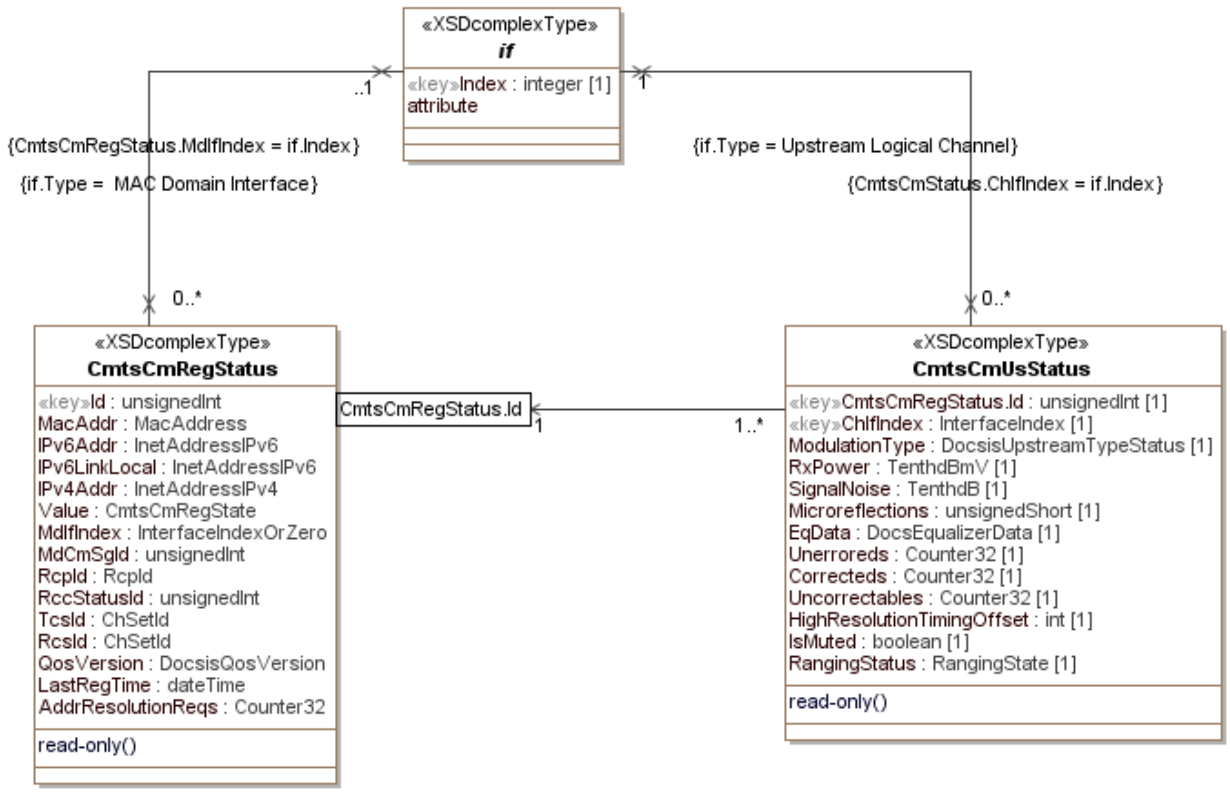


Figure 7.15: DOCS-IF3-MIB Performance Management Objects

7.2.2.3 DOCS-L2VPN-MIB Statistics Objects

The objects in the DOCS-L2VPN-MIB: Statistics Objects are taken from the DOCS-L2VPN-MIB specified in Annex A of ES 203 385 [3] and are used without modification for the CCAP.

Reference: ES 203 385 [3], DOCS-L2VPN-MIB.

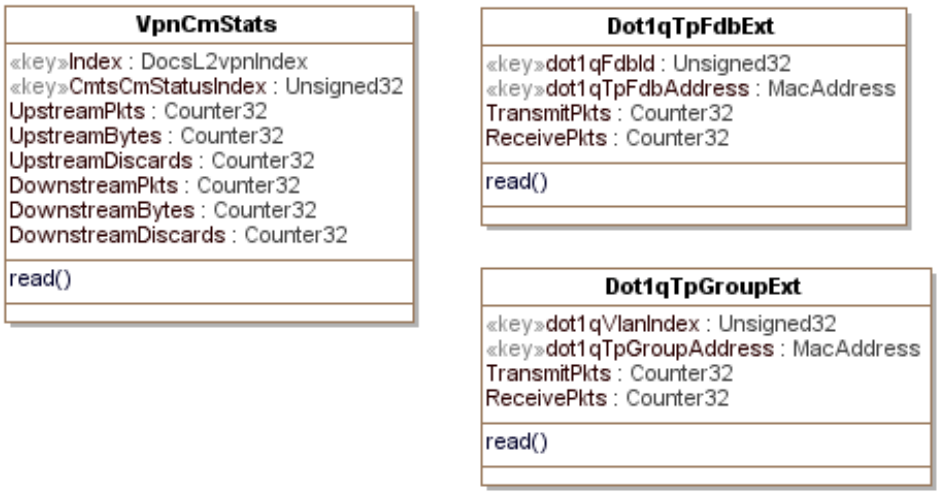


Figure 7.16: DOCS-L2VPN-MIB: Statistics Objects

7.2.2.4 DOCS-MCAST-MIB

The following objects in the DOCS-MCAST-MIB are taken from the DOCS-MCAST-MIB specified in Annex Q of [7] and used without modification for the CCAP:

- DsidPhs
- if
- CmtsReplSess

The docsBpi2CmtsIpMulticastMapTable object is taken from the DOCS-IETF-BPI2-MIB specified in RFC 4131 [i.13] and used without modification for the CCAP.

Reference: [7], DOCS-MCAST-MIB; RFC 4131 [i.13].

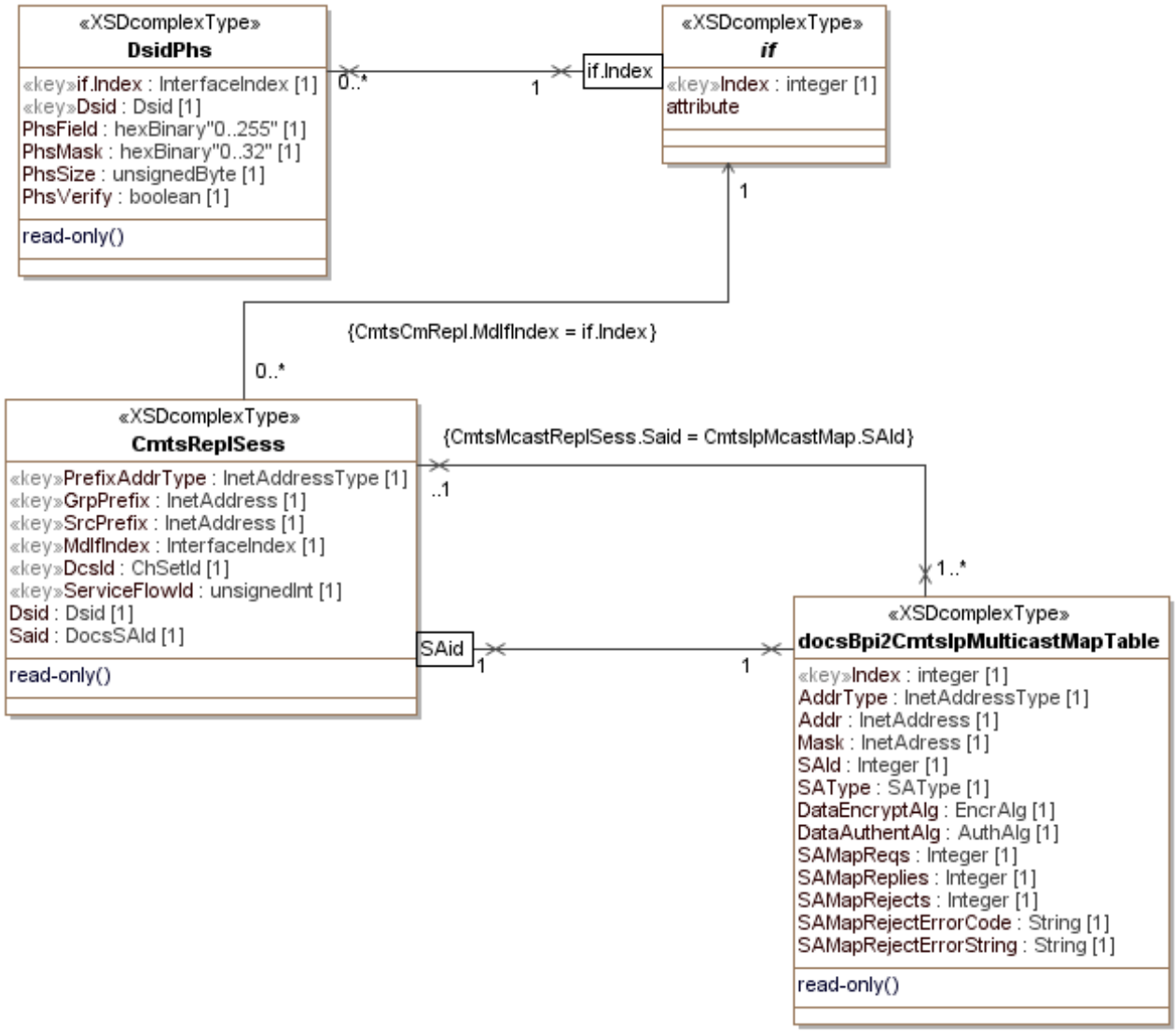


Figure 7.17: DOCS-MCAST-MIB Performance Management Objects

7.2.2.5 DOCS-QOS3-MIB: Statistical Objects

The objects in the DOCS-QOS3-MIB: Statistical Objects are taken from the DOCS-QOS3-MIB specified in Annex Q of [7] and used without modification for the CCAP.

Reference: [7], DOCS-QOS3-MIB.

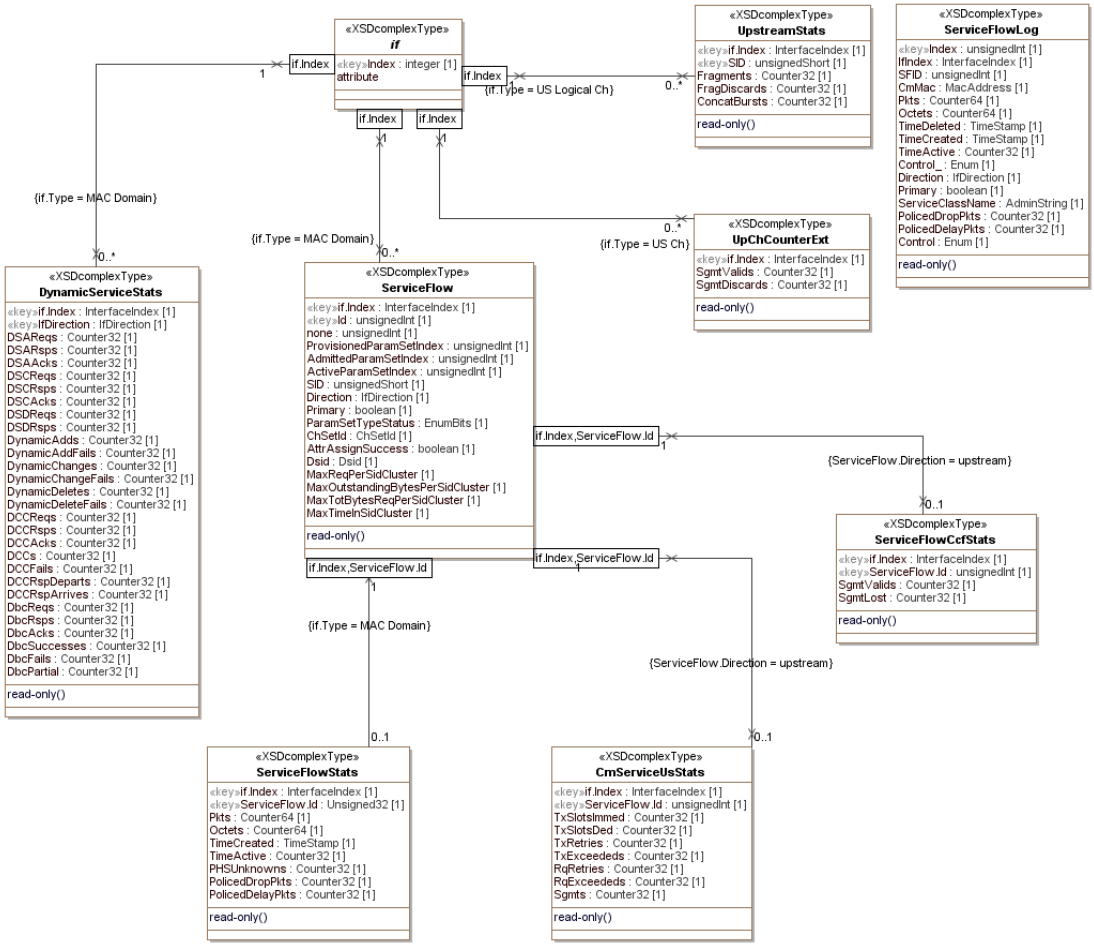


Figure 7.18: DOCS-QOS3-MIB: Statistical Objects Performance Management Objects

7.2.2.6 SCTE-HMS-MPEG-MIB: Statistics Objects

The objects in the SCTE-HMS-MPEG-MIB: Statistics Objects are taken from [29] and used with the following modifications for the CCAP.

The CcapMpegOutputProg object replaces the MpegOutputProg object from the SCTE-HMS-MPEG-MIB. It is defined in clause 7.2.1.10.5.

Reference: ANSI SCTE 154-4 [29].

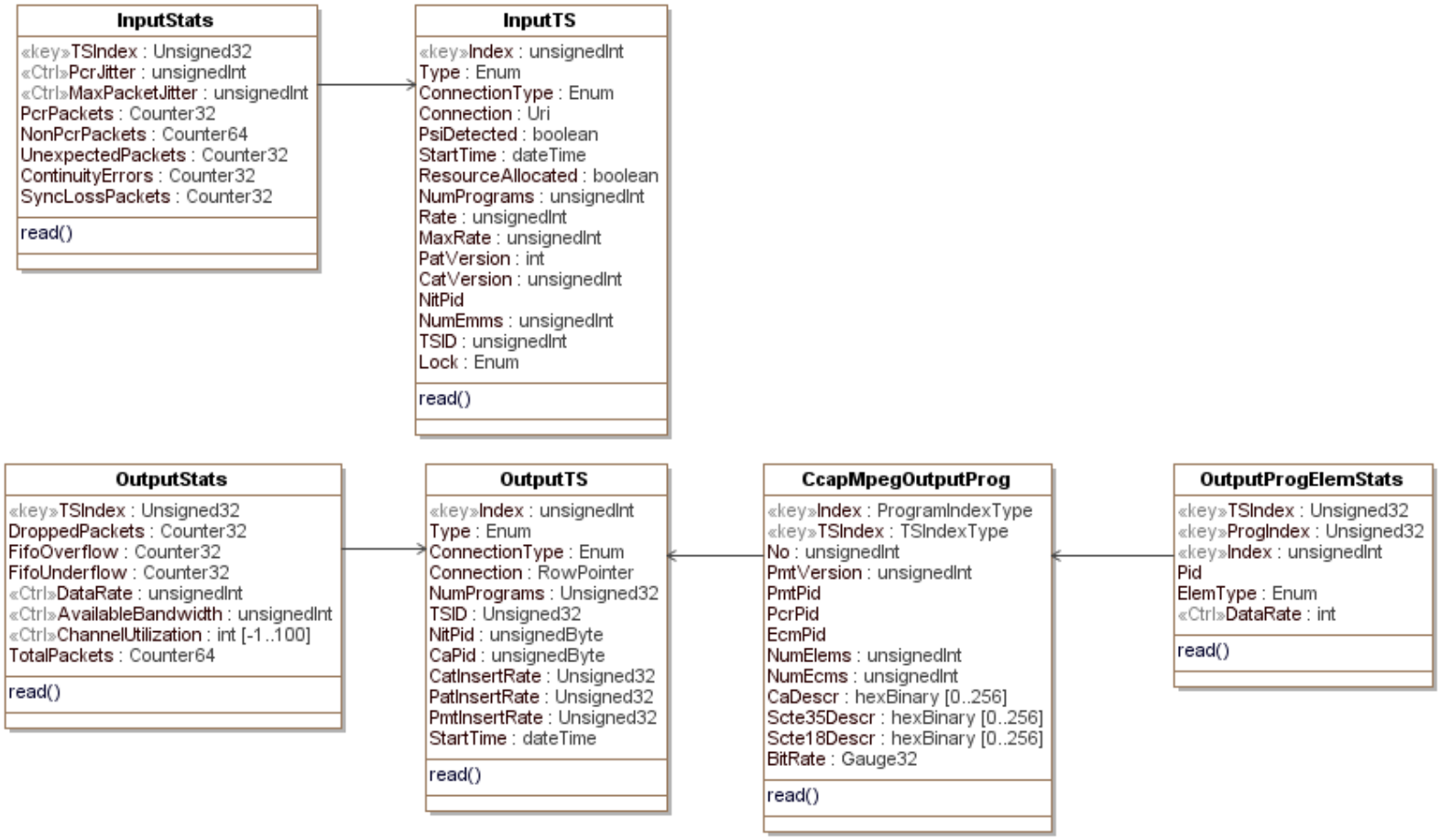


Figure 7.19: SCTE-HMS-MPEG-MIB: Statistics Objects Performance Management Objects

7.3 IPDR

The CCAP shall implement IPDR/SP as described in [7].

The CCAP shall support IPDR reporting on all of its access network interfaces (QAM, PON, etc.).

7.3.1 IPDR Service Definitions

The CCAP shall support all IPDR service definitions defined as mandatory in [7], including SAMIS. Additional service definitions may be identified in later versions of the present document. Refer to DOCSIS IPDR Service Definitions figure in [7] for the IPDR service definition object diagram.

If the CCAP supports PON interfaces, the CCAP shall support all IPDR service definitions defined as mandatory in [2].

8 Accounting Management

8.1 SAMIS

The CCAP shall support collection of usage information, for use by the billing system, via an interface known as the Subscriber Accounting Management Interface Specification (SAMIS).

9 Fault Management and Reporting Requirements

9.1 Fault Management Requirements and Transport Protocols

The CCAP shall conform to the fault management requirements specified in section 8 of [7], except as noted in the following clauses.

9.2 Event Reporting

The CCAP shall log events using standard mechanisms defined in section 8 of [7].

The CCAP shall support all Mandatory ("M") CMTS MIB objects that have an SNMP access type of accessible for SNMP Notifications ("Acc-FN") in Annex A of [7] and Annex A of ES 203 385 [3].

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

9.2.1 Event Notification

The 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 [21]).
2) Reported to SNMP entities as an SNMP notification.
3) Sent as a message to a Syslog server.

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 clause 6.5.10.6.4).

The CCAP shall support event notifications via local event logging.

The CCAP shall support event notifications via Syslog, including limiting/throttling, as specified in RFC 4639 [21].

The CCAP shall support event notification via SNMP traps, including limiting/throttling, as specified in RFC 4639 [21].

9.2.1.1 Format of Events

The following clauses explain in detail how the CCAP reports standard events by any of the following three mechanisms: local event logging, SNMP notification, and Syslog.

9.2.1.1.1 Local Event Logging

The CCAP shall maintain Local Log events, defined in RFC 4639 [21], in local non-volatile storage.

The CCAP may retain events designated for local volatile storage in local non-volatile storage.

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

Events are identical if their EventIds are identical. For identical events occurring consecutively, the 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 shall reflect the most recent event in the event description.

9.2.1.1.2 SNMP Traps

The CCAP shall implement the generic SNMP notifications according to Annex A of [7].

The CCAP shall implement SNMP notifications defined in DOCS-IF3-MIB from Annex A of [7].

The CCAP shall support at least 4 SNMP trap destinations.

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

9.2.1.1.3 Syslog

The CCAP shall support logging via Syslog per [7].

The CCAP shall support at least 4 Syslog servers as recipients.

The CCAP shall 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 shall send it in the following format:

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

Each of these fields is defined in the Syslog Message Format section of [7].

9.2.1.2 Standard Events for CCAP

The CCAP shall 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 clause 6.5.10.6.4).

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 shall report the event as a single event in the docsDevEventTable.

Event priority levels for the CCAP will use the following categories:

Emergency(1) events indicate fatal hardware or software failure that prevent normal system operation (all service 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 the present document (see clause 6.5.10.6.4).

9.2.2 Event Priorities and Vendor-Specific Events

The present document 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 9.1 summarizes those considerations.

Table 9.1: Event Priorities Assignment

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

9.2.3 NETCONF Notifications

NETCONF Notifications [24] 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, as specified in [24].

If the CCAP implements NETCONF Notifications towards OSS, the CCAP shall use the YANG module specified for this purpose in Annex E: YANG Module for Event Messaging (Normative).

9.3 Fault Management UML Object Model

9.3.1 Event Notification Objects

The objects for CCAP Event Notification are derived from the docsDevEventTable in RFC 4639 [21] and are used without modification. They are shown here for completeness.

Reference: RFC 4639 [21].

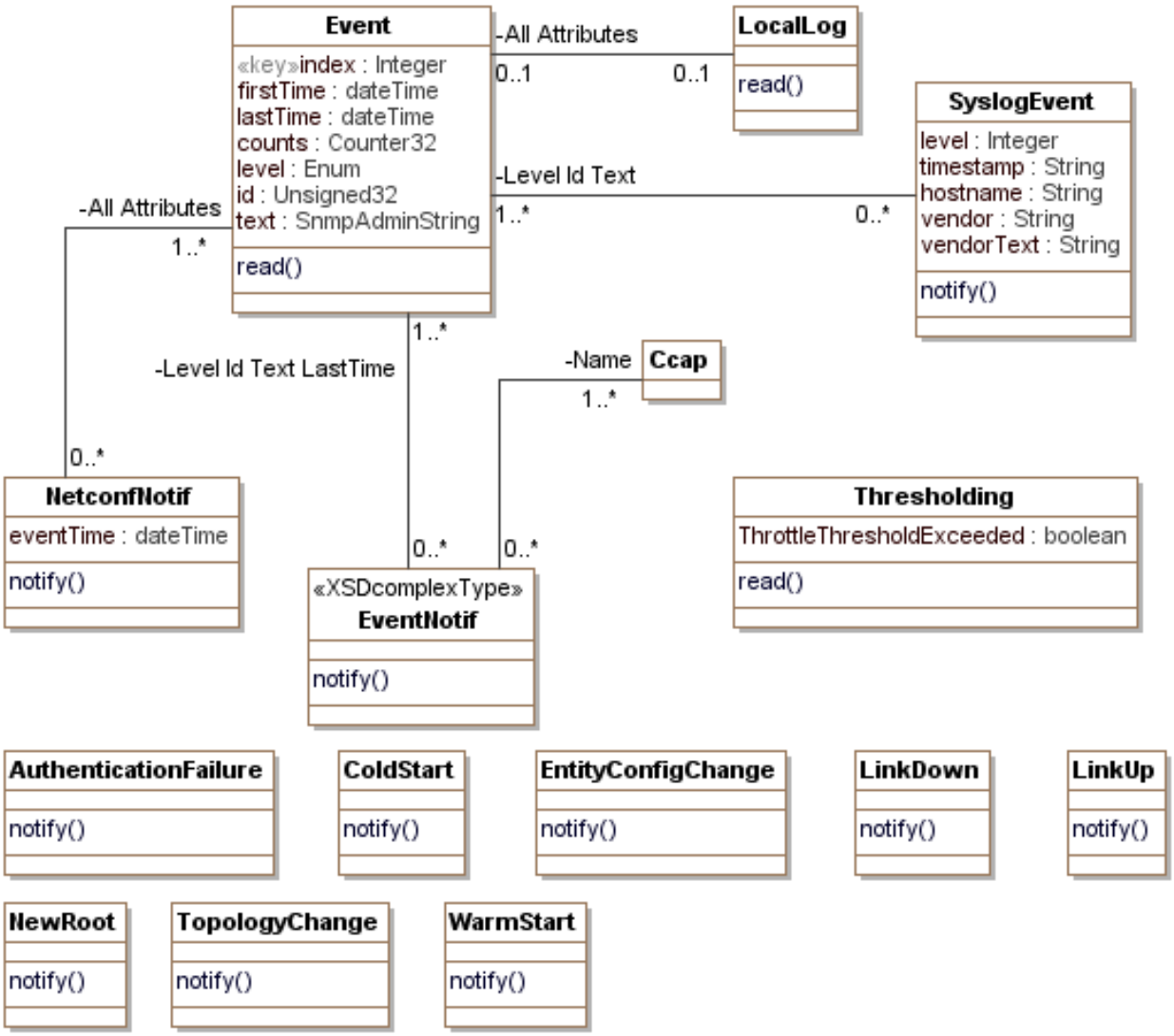
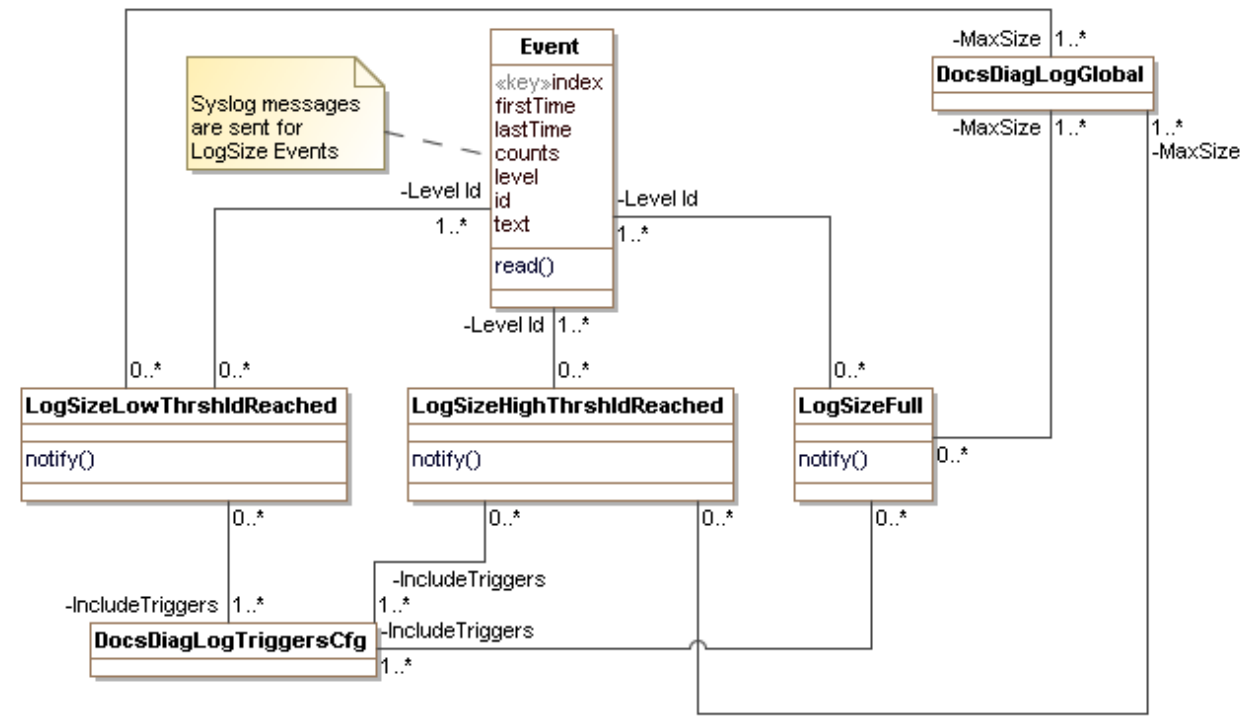


Figure 9.1: CCAP Event Notification Objects

9.3.2 CCAP CM Diagnostic Log Objects

These fault management objects are defined in [7] and will be used with no modifications for CCAP. They are shown here for completeness.

Reference: [7], DOCS-DIAG-MIB.



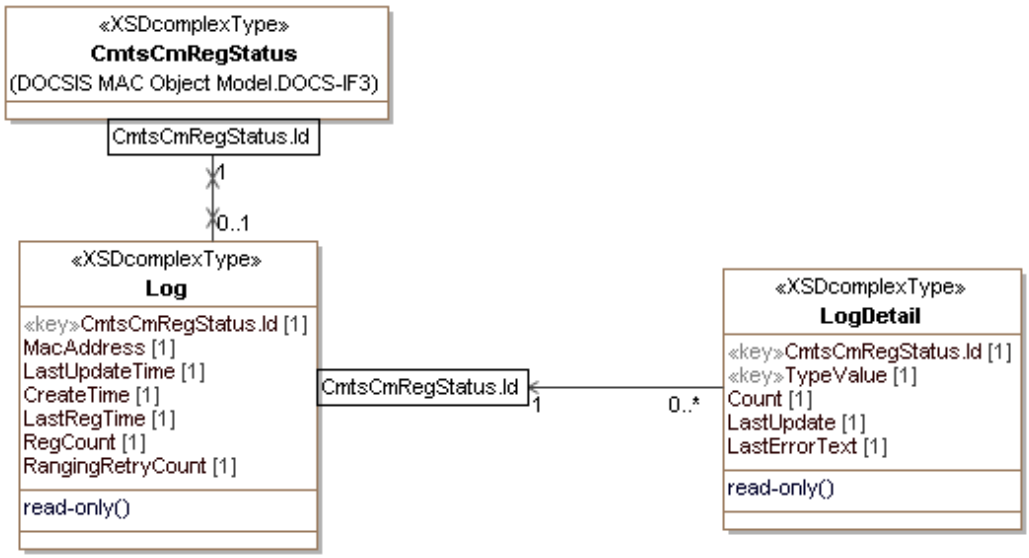


Figure 9.2: CCAP CM Diagnostic Log Objects

Annex A (normative): SNMP MIBs

A.1 CCAP MIB

The CCAP MIB is defined as follows:

```
CCAP-MIB DEFINITIONS ::= BEGIN
IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
    Unsigned32,
    Gauge32,
    Counter64
                        FROM SNMPv2-SMI
    TruthValue
                        FROM SNMPv2-TC
    OBJECT-GROUP,
    MODULE-COMPLIANCE
                        FROM SNMPv2-CONF
    SnmpAdminString
                        FROM SNMP-FRAMEWORK-MIB

    PhysicalIndexOrZero
                        FROM ENTITY-MIB
    mpegInputTSIndex,
    mpegInputProgIndex,
    mpegOutputTSIndex,
    mpegOutputProgIndex,
    mpegInputProgEntry,
    mpegOutputProgEntry,
    mpegVideoSessionIndex
                        FROM SCTE-HMS-MPEG-MIB
    ifIndex
                        FROM IF-MIB
    clabProjDocsis
                        FROM CLAB-DEF-MIB;

ccapMib MODULE-IDENTITY
    LAST-UPDATED "201304040000Z" -- April 04, 2013

    ORGANIZATION "Cable Television Laboratories, Inc."
    CONTACT-INFO
        "Postal: Cable Television Laboratories
         858 Coal Creek Circle
         Louisville, CO 80027-9750
         U.S.A.
         Phone: +1 303-661-9100
         Fax:   +1 303-661-9199
         E-mail:  mibs@cablelabs.com"
     DESCRIPTION
         "This MIB module contains additional management objects
         needed for the management of CCAP devices.
         Copyright 2011 Cable Television Laboratories, Inc.
         All rights reserved."
     REVISION "201304040000Z" -- April 04, 2013
     DESCRIPTION
         "Editorial correction of the letter case in the term XPath. Revised per
         ECN CCAP-OSSI-N-13.1095-2 and published as part of CM-SP-CCAP-OSSI-I04."
     REVISION "201208090000Z" -- August 9, 2012
     DESCRIPTION
         "Added an example of XPath expression containing interface name.
         Published as part of CM-SP-CCAP-OSSI-I03."
     REVISION "201108050000Z" -- August 5, 2011
     DESCRIPTION
         "Redefined ccapMpegDecryptSessionTable to allow it to compile properly.
          Removed ccapChassisMgmtObjects.
         Published as part of CM-SP-CCAP-OSSI-D03."
     REVISION "201105170000Z" -- May 17, 2011
     DESCRIPTION
         "Initial version, published as part of
```

```
          CM-SP-CCAP-OSSI-D01."
    ::= { clabProjDocsis 24 }

-- Textual Conventions

-- Object Definitions
ccapNotifications OBJECT IDENTIFIER ::= { ccapMib 0 }
ccapObjects OBJECT IDENTIFIER ::= { ccapMib 1 }

ccapInterfacesObjects OBJECT IDENTIFIER ::= { ccapObjects 1 }

ccapInterfaceIndexMapTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF CcapInterfaceIndexMapEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This object reports the correspondent device path for the
        Interface index defined by an object instance."
    ::= { ccapInterfacesObjects 1 }

ccapInterfaceIndexMapEntry    OBJECT-TYPE
    SYNTAX      CcapInterfaceIndexMapEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The conceptual row of ccapInterfaceIndexMapTable.
        The index of this object corresponds to the Interface MIB index
        for interfaces of IANA interface types:
         'docsCableMaclayer' -- '127'
         'docsCableDownstream' -- '128'
         'docsCableUpstream' -- '129'
         'docsCableUpstreamChannel' -- '205'
         'docsCableUpstreamRfPort' -- '256'
         'cableDownstreamRfPort' -- '257'"
    INDEX {
            ifIndex
          }
    ::= { ccapInterfaceIndexMapTable 1 }

CcapInterfaceIndexMapEntry ::= SEQUENCE {
    ccapInterfaceIndexMapPath
                 SnmpAdminString,
    ccapInterfaceIndexMapEntPhysicalIndex
                 PhysicalIndexOrZero

      }

ccapInterfaceIndexMapPath    OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This attribute indicates the CCAP node XPath expression that
         identifies the resource associated with the interface index.

         For example,
         The path value of the resource associated with an
         upstream logical channel with index =  5,
         in upstream physical channel index = 7,
         in an Upstream  RF port number = 15,
         from an US RF Line Card,
         in slot number = 3,
         chassis id = 1 is represented as:

         /ccap/chassis[id=1]
         /slot[number=3]
         /rf-line-card
         /us-rf-port[number=15]
         /upstream-physical-channel[index=7]
         /upstream-logical-channel[index=5]

         In another example, the path value of the resource associated with a
         DOCSIS MAC domain with name = 'ca5/1/0'is represented as:

         /ccap/docsis/docs-mac-domain/mac-domain[mac-domain-name='ca5/1/0']
         Note: Line breaks in the examples above were added for clarity."

    ::= { ccapInterfaceIndexMapEntry 1 }
```

```
ccapInterfaceIndexMapEntPhysicalIndex    OBJECT-TYPE
    SYNTAX      PhysicalIndexOrZero
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This Attribute corresponds to the entPhysical Index associated with
        the resource. Zero if undefined"
    ::= { ccapInterfaceIndexMapEntry 2 }

ccapMpegObjects OBJECT IDENTIFIER ::= { ccapObjects 2 }

ccapMpegInputProgTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF CcapMpegInputProgEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This object extends the SCTE-HMS-MPEG-MIB
         mpegInputProgTable for the CCAP system."
    ::= { ccapMpegObjects 1 }

ccapMpegInputProgEntry    OBJECT-TYPE
    SYNTAX      CcapMpegInputProgEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The Conceptual row of ccapMpegInputProgTable."
    AUGMENTS { mpegInputProgEntry }
    ::= { ccapMpegInputProgTable 1 }

CcapMpegInputProgEntry ::= SEQUENCE {
    ccapMpegInputProgBitRate
                 Gauge32,
    ccapMpegInputProgRequestedBandwidth
                 Unsigned32
      }

ccapMpegInputProgBitRate    OBJECT-TYPE
    SYNTAX      Gauge32
    UNITS       "bps"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Indicates the measured MPEG input program bitrate in bps."
    ::= { ccapMpegInputProgEntry 1 }

ccapMpegInputProgRequestedBandwidth    OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "bps"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Requested bandwidth for this MPEG input program.
        This value is used to validate the total QAM  bandwidth before
        allowing the creation of a new  session.

        It is also used to validate the input program  bandwidth
        overflow situation during the transmission.

        In the case of special stream without PCR, it is  used to limit
        the output bandwidth of that special program.

        A zero (0) value is returned if no bandwidth  validation is done
        on this program."
    ::= { ccapMpegInputProgEntry 2 }

ccapMpegOutputProgTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF CcapMpegOutputProgEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This object extends the SCTE-HMS-MPEG-MIB
        mpegOutputProgTable for the CCAP video
        down channel."
    ::= { ccapMpegObjects 2 }

ccapMpegOutputProgEntry    OBJECT-TYPE
```

```
    SYNTAX      CcapMpegOutputProgEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The Conceptual row of ccapMpegOutputProgTable."
    AUGMENTS { mpegOutputProgEntry }
    ::= { ccapMpegOutputProgTable 1 }

CcapMpegOutputProgEntry ::= SEQUENCE {
    ccapMpegOutputProgBitRate
                 Gauge32
      }

ccapMpegOutputProgBitRate    OBJECT-TYPE
    SYNTAX      Gauge32
    UNITS       "bps"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Indicates the output program bitrate in bps."
    ::= { ccapMpegOutputProgEntry 1 }

ccapMpegInputProgVideoSessionTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF CcapMpegInputProgVideoSessionEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This object specifies the list of video sessions
        that the MPEG input program are feeding."
    ::= { ccapMpegObjects 3 }

ccapMpegInputProgVideoSessionEntry    OBJECT-TYPE
    SYNTAX      CcapMpegInputProgVideoSessionEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The conceptual row of
        ccapMpegInputProgVideoSessionTable."
    INDEX {
            mpegInputTSIndex,
            mpegInputProgIndex,
            mpegVideoSessionIndex
          }
    ::= { ccapMpegInputProgVideoSessionTable 1 }

CcapMpegInputProgVideoSessionEntry ::= SEQUENCE {
    ccapMpegInputProgVideoSessionStatus
                 INTEGER
      }

ccapMpegInputProgVideoSessionStatus    OBJECT-TYPE
    SYNTAX      INTEGER {
                          active(1),
                          closed(2)
                        }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This attribute indicates the status of the session.
        Only active sessions need to be reported."
    ::= { ccapMpegInputProgVideoSessionEntry 1 }


ccapMpegOutputProgVideoSessionTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF CcapMpegOutputProgVideoSessionEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This object specifies the video sessions that are used
        to feed the video down channel program."
    ::= { ccapMpegObjects 4 }

ccapMpegOutputProgVideoSessionEntry    OBJECT-TYPE
    SYNTAX      CcapMpegOutputProgVideoSessionEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The conceptual row of ccapMpegOutputProgVideoSessionTable."
```

```
    INDEX {
            mpegOutputTSIndex,
            mpegOutputProgIndex,
            mpegVideoSessionIndex
          }
    ::= { ccapMpegOutputProgVideoSessionTable 1 }

CcapMpegOutputProgVideoSessionEntry ::= SEQUENCE {
    ccapMpegOutputProgVideoSessionStatus
                INTEGER
      }

ccapMpegOutputProgVideoSessionStatus    OBJECT-TYPE
    SYNTAX      INTEGER {
                          active(1),
                          closed(2)
                        }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This attribute indicates the status of the session.
        Only active sessions need to be reported."
    ::= { ccapMpegOutputProgVideoSessionEntry 4 }


ccapCryptoObjects OBJECT IDENTIFIER ::= { ccapObjects 3 }

ccapEcmgStatusTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF CcapEcmgStatusEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This object reports status information for each instance
        of an Entitlement Control Message Generator (ECMG)."
    ::= { ccapCryptoObjects 1 }

ccapEcmgStatusEntry    OBJECT-TYPE
    SYNTAX      CcapEcmgStatusEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The conceptual row of ccapEcmgStatusTable."
    INDEX {
            ccapEcmgIndex
          }
    ::= { ccapEcmgStatusTable 1 }

CcapEcmgStatusEntry ::= SEQUENCE {
    ccapEcmgIndex
                Unsigned32,
    ccapEcmgNumActiveSessions
                Gauge32,
    ccapEcmgCwMessageCount
                Counter64
    }

ccapEcmgIndex    OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This Attribute identifies an instance of an ECMG."
    ::= { ccapEcmgStatusEntry 1 }

ccapEcmgNumActiveSessions OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
    "The current number of encryption sessions managed by this ECMG."
    ::= { ccapEcmgStatusEntry 2 }

ccapEcmgCwMessageCount OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
    "The number of Control World (CW) messages received from this ECMG."
```

```
    ::= { ccapEcmgStatusEntry 3 }


ccapEcmdStatusTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF CcapEcmdStatusEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This object reports status information for each instance
        Entitlement Control Message Decoder (ECMD)."
    ::= { ccapCryptoObjects 2 }

ccapEcmdStatusEntry    OBJECT-TYPE
    SYNTAX      CcapEcmdStatusEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The conceptual row of ccapEcmdStatusTable."
    INDEX {
            ccapEcmdIndex
          }
    ::= { ccapEcmdStatusTable 1 }

CcapEcmdStatusEntry ::= SEQUENCE {
    ccapEcmdIndex
                 Unsigned32,
    ccapEcmdNumActiveSessions
                 Gauge32,
    ccapEcmdCwMessageCount
                 Counter64
      }

ccapEcmdIndex    OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This Attribute identifies an instance of ECMD."
    ::= { ccapEcmdStatusEntry 1 }

ccapEcmdNumActiveSessions OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
    "The current number of encryption sessions managed by this ECMD."
    ::= { ccapEcmdStatusEntry 2 }

ccapEcmdCwMessageCount OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
    "The number of Control World (CW) messages received from this ECMD."
    ::= { ccapEcmdStatusEntry 3 }

ccapMpegDecryptSessionTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF CcapMpegDecryptSessionEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The (conceptual) table of those video sessions defined in the
        mpegVideoSessionTable, from the SCTE-HMS-MPEG-MIB, which require
        decryption.

        Note that this table is potentially sparse: a (conceptual) entry
        exists only if the video session requires decryption."
    ::= { ccapCryptoObjects 3 }

ccapMpegDecryptSessionEntry    OBJECT-TYPE
    SYNTAX      CcapMpegDecryptSessionEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A (conceptual) entry for a decrypted video session.
        The mpegVideoSessionIndex represents the entry in the
        mpegVideoSessionTable that corresponds to the
        ccapMpegDecryptSessionEntry."
```

```
    INDEX {
            mpegVideoSessionIndex
           }
    ::= { ccapMpegDecryptSessionTable 1 }

CcapMpegDecryptSessionEntry ::= SEQUENCE {
    ccapMpegDecryptSessionDecrypted
        TruthValue
    }

ccapMpegDecryptSessionDecrypted    OBJECT-TYPE
    SYNTAX     TruthValue
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "Denotes whether or not the video session was decrypted.
        This object should always be set to 'true'."
    DEFVAL     { true }
    ::= { ccapMpegDecryptSessionEntry 1 }


-- Conformance Definitions
ccapMibConformance OBJECT IDENTIFIER ::= { ccapMib 2 }
ccapMibCompliances OBJECT IDENTIFIER ::= { ccapMibConformance 1 }
ccapMibGroups      OBJECT IDENTIFIER ::= { ccapMibConformance 2 }

ccapCompliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
         "The compliance statement for the."
    MODULE -- this MODULE
    MANDATORY-GROUPS {
            ccapInterfacesGroup,
            ccapMpegGroup,
            ccapCryptoGroup
            }
    ::= { ccapMibCompliances 1 }

ccapInterfacesGroup OBJECT-GROUP
    OBJECTS {
              ccapInterfaceIndexMapPath,
              ccapInterfaceIndexMapEntPhysicalIndex
    }
    STATUS      current
    DESCRIPTION
             "Objects implemented in the ccapInterfacesGroup."
    ::= { ccapMibGroups 1 }

ccapMpegGroup OBJECT-GROUP
    OBJECTS {
              ccapMpegInputProgBitRate,
              ccapMpegInputProgRequestedBandwidth,
              ccapMpegInputProgBitRate,
              ccapMpegInputProgVideoSessionStatus,
              ccapMpegOutputProgVideoSessionStatus,
              ccapMpegOutputProgBitRate
    }
    STATUS      current
    DESCRIPTION
             "Objects implemented in the ccapMpegGroup."
    ::= { ccapMibGroups 2 }

ccapCryptoGroup OBJECT-GROUP
    OBJECTS {
              ccapEcmgNumActiveSessions,
              ccapEcmgCwMessageCount,
              ccapEcmdNumActiveSessions,
              ccapEcmdCwMessageCount,
              ccapMpegDecryptSessionDecrypted
    }
    STATUS      current
    DESCRIPTION
             "Objects implemented in the ccapCryptoGroup."
    ::= { ccapMibGroups 3 }
END
```

Annex B (normative): Extending the Configuration Data Model

While the majority of the CCAP configuration data model is standardized in the XML schema and YANG module, it is anticipated that vendors will extend the configuration data model to support vendor-proprietary functionality. This annex summarizes the guidelines that should be followed when extending the configuration data model and provides examples of how the configuration data model can be extended in YANG and in XML.

B.1 XML Schema Extension

Vendor-specific extensions to the CCAP XML schema are only allowed within the provided "<ext>" elements in the CCAP schema. Those extensions are proprietary to the vendor. The proprietary content is not defined within the present document.

Vendor-proprietary schemas intended to extend the standard XML schema are required to use a vendor-specific, globally-unique URI for the XML namespace for that vendor. Namespace URIs need to be chosen such that they cannot collide with standard or other enterprise namespaces; for example, the enterprise or organization name could be used in the namespace.

The CCAP XML schema provides a complex type that allows vendors to add a standardized version number to their vendor-specific extension. This complex data type is shown in Annex J: Vendor Schema Version in the CCAP XSD. While it is not mandatory for a vendor-specific extension to include a vendor version number, if a vendor version number is included, this complex type is required to be used to convey the version information. This version number is in addition to the CCAP XSD version information.

In addition to extension of the XML schema via complementary vendor-proprietary elements inserted within <ext> elements of the standard schema, a mechanism has been defined whereby vendors can extend their configuration model in YANG, but convert these extensions to XML schema. Refer to clause B.2 for details. In this case, a single XML configuration file will validate against both the vendor-proprietary XML schema and the standard schema.

The CCAP shall reject any XML configuration that would not validate against the standard XML schema.

B.1.1 Sample Vendor-Specific XSD Extensions

B.1.1.1 Extending a Standard Configuration Object

The following example adds attributes to the rf-line-card element in the XSD using both a simple type and a new, named complex type. This example vendor XSD file is referenced within configurations that include these new elements.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="urn:cablelabs:params:xml:ns:yang:vendor" xmlns:vendor="vendor"
xmlns:ccap="urn:cablelabs:params:xml:ns:yang:ccap" targetNamespace="vendor"
attributeFormDefault="unqualified" version="0001:000A" xml:lang="en">
  <xs:import namespace="urn:cablelabs:params:xml:ns:yang:ccap"
schemaLocation=" ccap@2013-04-04.xsd"/>
  <xs:element name="ds-annex" type="ccap:downstream-phy-type">
    <xs:annotation>
      <xs:documentation>Annex for entire DS RF card</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="rf-linecard-details" type="vendor:Rf-Card-Model">
    <xs:annotation>
      <xs:documentation>Type of line-card</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="Rf-Card-Model">
    <xs:sequence>
      <xs:element name="model" minOccurs="1">
        <xs:annotation>
          <xs:documentation> Model number of linecard</xs:documentation>
```

```
        </xs:annotation>
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="U1"/>
            <xs:enumeration value="D1"/>
            <xs:enumeration value="D2"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="num-rf-ports" type="xs:unsignedByte" minOccurs="1">
        <xs:annotation>
          <xs:documentation>Maximum number of RF ports on the card</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

B.1.1.1.1 Sample Configuration File Using Extended Standard Configuration Objects

In the following example, the vendor-proprietary XSD from the previous section is used to validate the following XML configuration file.

```
<ccap:ccap xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" SchemaVersion="2013-04-04"
xsi:schemaLocation="urn:cablelabs:params:xml:ns:yang:ccap ccap@2013-04-04.xsd"
xmlns:ccap="urn:cablelabs:params:xml:ns:yang:ccap" operation="merge">
  <chassis>
    <slot>
      <slot-number>1</slot-number>
      <rf-line-card>
        <rf-card>
        <line-card-name>Downstream RF Line Card 1</line-card-name>
        <admin-state>up</admin-state>
        <protected-by>2</protected-by>
        </rf-card>
        <encryptor>
          <encryptor-index>1</encryptor-index>
          <ca-encryptor-type>motorola</ca-encryptor-type>
          <ecm-timeout>10</ecm-timeout>
          <clear-stream-timeout>10</clear-stream-timeout>
          <ecmg-usage>
            <ecmg-usage-index>1</ecmg-usage-index>
            <priority>1</priority>
            <ecmg-ref>1</ecmg-ref>
          </ecmg-usage>
        </encryptor>
        <ext>
          <vendor:ds-annex xsi:schemaLocation="vendor vendor.xsd" xmlns:vendor="vendor"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">j83annexB</vendor:ds-annex>
          <vendor:rf-linecard-details xmlns:vendor="vendor"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <model>D2</model>
            <num-rf-ports>12</num-rf-ports>
          </vendor:rf-linecard-details>
        </ext>
      </rf-line-card>
    </slot>
  </chassis>
</ccap:ccap>
```

B.1.1.2 Extending by Adding a New Object Type

The following example defines an XSD schema for a new vendor-specific configuration object, realizing the CLI command "crypto pki token default removal timeout [*seconds*]".

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.vendor2.com/example-ns-
partial-crypto-ccap" xmlns:ccap="urn:cablelabs:params:xml:ns:yang:ccap" xmlns:ns1="vendor2"
targetNamespace="vendor2" elementFormDefault="qualified" attributeFormDefault="unqualified"
version="2010-11-8" xml:lang="en">
  <xs:import namespace="urn:cablelabs:params:xml:ns:yang:ccap" schemaLocation="ccap@2013-04-
04.xsd"/>
  <xs:annotation>
    <xs:documentation xml:lang="en">
      An example of a schema defining the crypto CLI command.
    </xs:documentation>
```

```
  </xs:annotation>
  <xs:element name="crypto">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="pki">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="token">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="default">
                       <xs:complexType>
                         <xs:sequence>
                           <xs:element name="removal">
                              <xs:complexType>
                                <xs:sequence>
                                  <xs:element name="timeout">
                                    <xs:complexType>
                                      <xs:sequence>
                                  <xs:element name="TokenKeyTimeoutSeconds" type="xs:unsignedInt"/>
                                       </xs:sequence>
                                    </xs:complexType>
                                  </xs:element>
                                </xs:sequence>
                              </xs:complexType>
                           </xs:element>
                         </xs:sequence>
                       </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

B.1.1.2.1 Sample Configuration File Using the New Vendor Extension Objects

In the following example, the vendor-proprietary XSD from the previous section is used to validate the following XML configuration file.

```
<ccap:ccap xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" SchemaVersion="2013-04-04"
xmlns:ccap="urn:cablelabs:params:xml:ns:yang:ccap" operation="merge" xsi:schemaLocation="vendor2
CCAP-vendor-extension-example-2.xsd">
  <ext>
    <vendor-extension-version  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <major-version>1</major-version>
      <minor-version>0</minor-version>
    </vendor-extension-version>
    <crypto xsi:schemaLocation="vendor2 CCAP-vendor-extension-example-2.xsd"
xmlns="http://www.vendor2.com/example-ns-partial-crypto-ccap">
      <pki>
        <token>
          <default>
            <removal>
              <timeout>
                <TokenKeyTimeoutSeconds>11</TokenKeyTimeoutSeconds>
              </timeout>
            </removal>
          </default>
        </token>
      </pki>
    </crypto>
  </ext>
</ccap:ccap>
```

NOTE: The above example of vendor extension schema has been developed using design principles diverging from the object oriented methodology utilized throughout the present document. The goal of such an example is to demonstrate the flexibility in defining vendors extensions. The example should not be considered a methodology or a style recommendation.

B.2 YANG Configuration Model Extension

Any extensions to the YANG configuration data model are required to adhere to the requirements in clause 6.5.2.

B.2.1 YANG Extension Principles

Extensions to the YANG configuration data structure are required to be defined in a separate module, rather than within one of the standard CCAP module files. Doing so leaves the standard configuration object model intact and helps to ensure interoperability.

Vendor-proprietary sub-node extensions to standard "list", "choice", and "container" elements are permitted via the use of the "augment" syntax within the vendor-proprietary YANG module. These extensions are only allowed to the "yang-ext" container (which is included in elements eligible for extension). This requirement is to ensure that when the vendor-proprietary YANG module (which imports the standard module) is converted to XML schema, that instance documents valid against the resulting schema are also valid against the standard schema.

In general, vendor-proprietary extensions to the standard YANG module should not use "deviation" statements to alter standard configuration objects. As the fundamental requirement is that nothing be done via YANG extension that would cause configurations valid against the vendor's XML schema to be invalid against the standard schema, deviations are only viable when they place tighter restrictions on an element than the standard schema does.

Vendor-proprietary extensions to the standard YANG modules are required to use a vendor-specific, globally-unique URI for the XML namespace for that vendor. Namespace URIs are chosen so that they cannot collide with standard or other enterprise namespaces; for example the enterprise or organization name could be used in the namespace.

B.2.2 Creating Vendor Extensions

This section provides a few illustrative examples of creating vendor extensions in YANG. Refer to RFC 4254 [i.19] for a complete reference to the extension mechanisms of the YANG language.

B.2.2.1 Specifying the Vendor-Proprietary Namespace in YANG

When creating a vendor-specific YANG extension file, the vendor's namespace is required. Vendors that intend to extend the standard YANG module will use a unique URI to define the XML namespace. The following example depicts this concept.

```
module example-ccap-extension {
    yang-version 1;
    namespace "http://www.example.com/ccap-extension";
    prefix "vendor-ext";
    import ccap { prefix "ccap";  revision-date "2012-04-01";  }
    organization "EXAMPLE VENDOR";
    contact
        "WG-email:  example@vendor.com";
    description
        "Vendor Specific";
    revision "2012-04-01" {
        description "Initial version ";
    }

    container ccap {
        uses ccap:ccap-group;
    }
} //  vendor-module
```

B.2.2.2 Extending a Container or List in YANG

To extend standard configuration objects with vendor-proprietary objects, the "augment" syntax is used to define the location where new nodes are inserted into the standard YANG module, as well as to define the new nodes to be inserted. An "augment" statement always adds a new node to the configuration model and is only allowed, per the present document, in the "yang-ext" elements that are provided in the standard YANG module precisely for this purpose.

Note that using the "deviation" syntax to extend the YANG configuration data model is only allowed in the cases outlined below.

Tables B.1 and B.2 summarize the acceptable ways to extend CCAP configuration data model objects.

Table B.1: Extending CCAP Configuration Objects with the "augment" Statement

Object	Extension Use Case	Method to Extend

Container	Add new data node (leaf, list, etc.) to container	Augment the container with new data node. The following example adds a new leaf to the chassis container.
		
		`augment "/ccap:ccap/ccap:chassis/yang-ext" {`
		`  leaf contact-name {`
		`  type string;`
		`  description "Contact name";`
		`  }`
		`}`

List	Add new data node (leaf, list, etc.) to list	Augment a list with new data node. The following example adds a new leaf to the ds-rf-port -group object.
		
		`augment "/ccap:ccap/ccap:chassis/ccap:slot/ccap:line-card-type/ccap:rf-line-card/ccap:rf-line-card/ccap:ds-rf-port/yang-ext" {`
		`  leaf super-spectrum {`
		`  type  boolean;`
		`  mandatory  false;`
		`  description "Turns on or off the super spectrum feature.";`
		`  }`
		`}`

Choice	Add a new case to an existing choice object	Augment a choice with a new case data definition. The following example adds a new line card type to the line-card choice node.
		
		`augment "/ccap:ccap/ccap:chassis/ccap:slot/ccap:line-card-type/yang-ext/yang-choice-ext" {`
		`  case vendor-new-line-card {`
		`  list rf-port {`
		`  key "port-number";`
		`  uses ccap:port-group;`
		`  }`
		`  }`
		`}`

type	Change the range attribute associated with a typedef	The range specified for an existing typedef can be altered when included in a new leaf, as long as the new range specified is more narrow than the default range. The following example sets a smaller range for the InetPortNumber typedef when used in the new port-number leaf.
		
		`augment "/ccap:ccap/ccap:chassis/ccap:slot/ccap:line-card-type/ccap:rf-line-card/ccap:rf-line-card/yang-ext" {`
		`  leaf admin-port-number {`
		`  type inet:port-number {`
		`  range "1..45";`
		`  }`
		`}`

Table B.2: Extending CCAP Configuration Objects with the "deviation" Statement

Extension Use Case	Method to Extend

Add a range where one did not exist or replace an existing range	`deviation "/ccap:ccap/ccap:chassis/ccap:slot/ccap:slot-number" {`
	`  deviate replace {`
	`  type int8 {`
	`  range "0..13";`
	`  }`
	`  }`
	`}`

Put a bound on the total number of items supported, where no max-elements existed	`deviation "/ccap:ccap/ccap:docsis/ccap:docs-mac-domain/ccap:mac-domain " {`
	`  deviate add {`
	`  max-elements 100;`
	`  }`
	`}`

Replace the bound on the total number of items supported, where a max-elements definition existed	`deviation "/ccap:ccap/ccap:docsis/ccap: docs-mac-domain/ccap:mac-domain" {`
	`  deviate replace {`
	`  max-elements 100;`
	`  }`
	`}`

Remove a default value and make the item mandatory	`// First remove the default`
	
	`deviation "/ccap:ccap/ccap:docsis/ccap:docs-mac-domain/ccap:mac-domain/ccap:mdd-interval" {`
	`  deviate delete {`
	`  default "2000";`
	`  }`
	`}`
	`//Now add the mandatory true property`
	`deviation "/ccap:ccap/ccap:docsis/ccap:docs-mac-domain/ccap:mac-domain/ccap:mdd-interval" {`
	`  deviate add {`
	`  mandatory "true";`
	`  }`
	`}`

B.2.3 Example Vendor-Proprietary Extensions in YANG Configuration Messages

The following examples show a vendor-extension YANG module and a partial CCAP configuration that uses those vendor extensions.

B.2.3.1 Sample Vendor-Extension YANG Module

In this example, the following elements are extended:

- A contact-name leaf is added to the chassis container.
- The InetPortNumber typedef has its range narrowed in the port-number leaf.
- A new case is added to the card-type choice definition.

```
module example-ccap-extension {
    yang-version 1;
    namespace "http://www.example.com/ccap-extension";
    prefix "ccap-extension";
    import ccap                 { prefix "ccap";    }
    organization
        "Example Vendor";
    contact
        "WG-email:  example@vendor.com";
    description
        "Vendor Specific";
```

```
    revision "2013-04-04" {
        description "Initial version ";
    }
  augment "/ccap/chassis/slot/line-card-type/rf-line-card/yang-ext1"  {
      leaf admin-port-number {
          type inet:port-number {
              range "1..45";
          }
      }
  }
augment "/ccap/chassis/slot/line-card-type/yang-choice-ext" {
        choice vendor-line-card {
            case vendor-turbo-card {
               list rf-port {
                 key "port-number";
                 uses ccap:port-group;
               }
            }
        }
  }
augment "/ccap/chassis/yang-ext" {
        leaf contact-name {
           type string;
           description "Contact name";
        }
  }
```

B.2.3.2 Sample Partial Configuration Message Using Vendor Extensions

```
<ccap:ccap nc:operation="merge" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  SchemaVersion="2013-04-04"
  xsi:schemaLocation="urn:arris:ns:yang:1.0:vendor ccap-vendor-yang-ext.xsd"
            xmlns:ccap="urn:arris:ns:yang:1.0:vendor">
<chassis>
    <slot>
        <slot-number>3</slot-number>
        <yang-choice-ext>
            <vendor-turbo-card>
                <rf-port>
                    <port-number>6</port-number>
                </rf-port>
            </vendor-turbo-card>
        </yang-choice-ext>
    </slot>
    <slot>
        <slot-number>4</slot-number>
        <rf-line-card>
             <yang-ext1>
                  <admin-port-number>27</admin-port-number>
             </yang-ext1>
        </rf-line-card>
    </slot
    <yang-ext>
        <contact-name>customer support</contact-name>
    </yang-ext>
  </chassis>
</ccap>
```

Annex C (normative): Format and Content for Event, Syslog, and SNMP Notification

This annex is an extension of the Format and Content for Event, SYSLOG, and SNMP Notification Annex of [7].

The CCAP shall support all mandatory CMTS events as defined in [7], as well as the list of events defined in table C.1.

Table C.1: CCAP Events

Process	Sub-Process	CCAP Priority	Event Message	Message Notes and Details	Error Code Set	Event ID	Trap Name
CCAP XML Configuration File Processing							
CCAP-Config	Login	Error	Inbound interactive login failed: Protocol: <P1>, Username: <P2>	P1=Protocol from IntegratedServers ServerType attribute (clause 6.5.8.5) P2=Username	F001.1	70000101	docsIf3CmtsEventNotif
CCAP-Config	File Transfer	Error	File transfer failed: Protocol: <P1>, Username: <P2>, Destination host/path: <P3>:<P4>	P1=Protocol from clause 6.3.7 File Transfer Mechanisms P2=Username P3=Destination host name or IP address P4=Path to filename	F001.2	70000102	docsIf3CmtsEventNotif
CCAP-Config	Validate	Info	XML Configuration File - Validation Passed: <P1>	P1=configuration file name	F001.3	70000103	docsIf3CmtsEventNotif
CCAP-Config	Validate	Notice	XML Configuration File - Validation Failed: <P1>	P1=configuration file name	F001.4	70000104	docsIf3CmtsEventNotif
CCAP-Config	Execute	Notice	XML Configuration File - Execution Success: <P1>	P1=configuration file name	F001.5	70000105	docsIf3CmtsEventNotif
CCAP-Config	Execute	Error	XML Configuration File - Unsupported Elements - Configuration Continued: <P1>	P1=configuration file name	F001.6	70000106	docsIf3CmtsEventNotif
CCAP-Config	Execute	Error	XML Configuration File - Non-fatal Error - Configuration Continued: <P1>	P1=configuration file name	F001.7	70000107	docsIf3CmtsEventNotif
CCAP-Config	Execute	Warning	XML Configuration File - Fatal Operation Value Error - Configuration Aborted: <P1>	P1=configuration file name	F001.8	70000108	docsIf3CmtsEventNotif
CCAP-Config	Execute	Warning	XML Configuration File - Fatal Error - Configuration Aborted: <P1>; <P2>	P1=configuration file name P2=error description	F001.9	70000109	docsIf3CmtsEventNotif
CCAP ERMI							
CCAP-ERMI		Critical	Session Loss type=<P1>; sessionId = <P2>;	P1 = session loss type P2 = sessionID	F002.1	70000201	docsIf3CmtsEventNotif

Process	Sub-Process	CCAP Priority	Event Message	Message Notes and Details	Error Code Set	Event ID	Trap Name
CCAP-ERMI		Critical	Link Down Loss of Service;Interface=<P1>;	for syslog & local-log Mandatory Add: ; Error Code = 0; P1= MapPath	F002.2	70000202	docsIf3CmtsEventNotif
CCAP-ERMI		Critical	Sessions Lost=<P1>; Sessions failed-over=<P2>	P1 = number of sessions lost P2 = number of failed-over sessions for syslog & local-log Mandatory Add: ; Error Code = 0;	F002.3	70000203	docsIf3CmtsEventNotif
CCAP-ERMI		Critical	Excessive network jitter in session, jitter buffer overflow; sessionID=<P1>	P1 = sessionID for syslog & local-log Mandatory Add: ; Error Code = 0;	F002.4	70000204	docsIf3CmtsEventNotif
CCAP Physical & Environmental							
CCAP- PE	Cooling	Critical	Cooling - Fan unit <P1> Failure; <P2>	P1 = entPhysicalIndex of fan unit P2 = entPhysicalName	F003.1	70000301	docsIf3CmtsEventNotif
CCAP-PE	Cooling	Warning	Cooling - Sensor unit=<P1> - High Temperature Threshold Exceeded <P2>	P1 = entPhysicalIndex of temperature sensor P2 = Temp (F/C)	F003.2	70000302	docsIf3CmtsEventNotif
CCAP-PE	Cooling	Warning	Cooling - Sensor unit=<P1> - Normal Operating Temperature Exceeded: <P2>	P1 = entPhysicalIndex of temperature sensor P2 = Temp (F/C)	F003.3	70000303	docsIf3CmtsEventNotif
CCAP-PE	Power	Critical	Power - Power Supply unit-<P1> - Bus Failure	P1 = entPhysicalIndex of power supply unit	F003.4	70000304	docsIf3CmtsEventNotif
CCAP-PE	Power	Warning	Power - Power supply unit=<P1>: <P2> - Below 95%	P1= entPhysicalIndex of power supply unit P2 = entPhysicalName of power supply unit	F003.5	70000305	docsIf3CmtsEventNotif

Process	Sub-Process	CCAP Priority	Event Message	Message Notes and Details	Error Code Set	Event ID	Trap Name
CCAP-PE	Power	Notice	Power - Power Supply Switchover, Previous unit=<P1>: <P2>, New unit=<P2>: <P4>	P1 = entPhysicalIndex of power supply unit P2 = entPhysicalName of power supply unit P3 = entPhysicalIndex of power supply unit P4 = entPhysicalName of power supply unit	F003.6	70000306	docsIf3CmtsEventNotif
CCAP-PE	Power	Critical	Power - Power Supply unit=<P1>: <P2> - Improper Input Voltage	P1 = entPhysicalIndex of power supply unit P2 = entPhysicalName of power supply unit	F003.7	70000307	docsIf3CmtsEventNotif
CCAP-PE	Power	Critical	Power - - Power Supply unit=<P1>: <P2> - Power Phase Disconnected	P1= entPhysicalIndex of power supply unit P2 = entPhysicalName of power supply unit e For Syslog and Local Log, append: CCAP shut down due to multiphase power problem	F003.8	70000308	docsIf3CmtsEventNotif
CCAP-PE	Power	Notice	Power - Power Supply unit=<P1>: <P2>; Operational	P1 = entPhysicalIndex of power supply unit P2 = entPhysicalName of power supply unit	F003.9	70000309	docsIf3CmtsEventNotif
CCAP-PE	Redundancy	Alert	Line Card Failure in slot=<P1> - No Redundancy	P1 = entPhysicalIndex of the slot number	F003.10	70000310	docsDevCmtsEventNotif
CCAP-PE	Redundancy	Critical	Line Card Failure in slot=<P1> failed over to redundant card in slot=<P2>	P1 = entPhysicalIndex of slot number of the failed line card P2 = entPhysicalIndex of slot number of the redundant line card	F003.11	70000311	docsDevCmtsEventNotif
CCAP-PE	Redundancy	Notice	Line Card Operational in slot=<P1>	<P1>=entPhysicalIndex of slot number	F003.12	70000312	docsDevCmtsEventNotif
CCAP-PE	Interface Status	Critical	Failover of interface ifIndex=<P1>, ifAlias=<P2> to interface ifIndex=<P3>, ifAlias<P4>	P1/P3 = ifIndex from ifTable for Ethernet Interface P2/P4 = ifAlias from ifTable for Ethernet Interface	F003.13	70000313	docsIf3CmtsEventNotif

Process	Sub-Process	CCAP Priority	Event Message	Message Notes and Details	Error Code Set	Event ID	Trap Name
CCAP-PE	Interface Status	Notice	Interface ifIndex=<P1>, ifAlias=<P2> Operational	P1 = ifIndex from ifTable for Ethernet Interface P2 = ifAlias from ifTable for Ethernet Interface	F003.14	70000314	docsIf3CmtsEventNotif
CCAP COPS Interface							
CCAP-COPS	Status	Critical	COPS Connection Limit Threshold Exceeded <TAGS>		F004.1	70000401	docsIf3CmtsEventNotif
CCAP Content Protection							
CCAP-CP	Encryptor	Alert	Stream not Restored; Manual intervention required: video traffic sessionId = <P1>	P1 = Video sessionId	F005.1	70000501	docsIf3CmtsEventNotif
CCAP Denial of Service Protection							
CCAP-DOS	Traffic	Error	Protocol throttling initiated: <P1>	P1 = Protocol being throttled	F006.1	70000601	docsIf3CmtsEventNotif

C.1 Example SNMP Notification and Syslog Event Message

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/;
```

Annex D (normative): CCAP Data Type Definitions

D.1 Overview

This annex includes the data type definitions for the object models defined for use in the CCAP. The Unified Modeling Language (UML) is used for modeling the management requirements. The data types defined in this annex are mapped for use with YANG data types.

D.2 Primitive Data Types

Table D.1 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 appropriated type in YANG. The YANG Built-In Data Type Mapping references YANG data types defined in RFC 6021 [i.20] or as described in table D.1.

Table D.1: 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

D.3 Derived Data Types

Table D.2 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 appropriated type in YANG. The YANG Derived Data Type Mapping references YANG data types defined in RFC 6021 [i.20] or as described in table D.2.

Table D.2: Derived Data Types

UML Derived Data Type	YANG Derived Data Type Mapping	Permitted Values
Counter32	counter32	
Counter64	counter64	
Gauge32	gauge32	
Gauge64	gauge64	
TimeTicks	timeticks	
TimeStamp	timestamp	
PhysAddress	phys-address	

UML Derived Data Type	YANG Derived Data Type Mapping	Permitted Values
MacAddress	mac-address	e.g. 01:23:45:67:89:ab
InetPortNumber	port-number	0..65 535
IPAddress	ip-address	IPv4 or IPv6 Address
IPv4Address	ipv4-address	IPv4 Address
IPv6Address	ipv6-address	IPv6 Address
InetAddressPrefixLength	address-prefix-len-type	0..2 040
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	

Annex E (normative): YANG Module for Event Messaging

Clause 9.2.3 NETCONF Notifications, allows NETCONF Notifications (specified in [24]) to be used for NETCONF-based notifications towards an OSS. IF NETCONF Notifications are being used, the CCAP will use the following YANG module for event messaging from the CCAP to an OSS.

```
module ccap-events {
  namespace "urn:cablelabs:params:xml:ns:yang:ccap:events";
  prefix "events";
  organization "Cable Television Laboratories, Inc.";
  contact
    "Postal: Cable Television Laboratories, Inc.
     858 Coal Creek Circle
     Louisville, Colorado 80027-9750
     U.S.A.
     Phone: +1 303-661-9100
     Fax:   +1 303-661-9199
     E-mail: mibs@cablelabs.com";
  reference
    "RFC 4639";
  revision 2011-03-22 {
    description
      "Initial revision.";
  }
  notification ccap-event {
    leaf sequence-number {
      type uint64;
      description
        "This is an identifier for a single event. It is expected to be unique
         across all streams ";
    }
    leaf level {
      type enumeration {
        enum emergency {
          value 1;
          description
            "Indicate vendor-specific fatal hardware or software errors that prevent
             normal system operation";
        }
        enum alert {
          value 2;
          description
            "Indicate a serious failure that causes the reporting system to reboot
             but that is not caused by hardware or software
             malfunctioning.";
        }
        enum critical {
          value 3;
          description
            "Indicate a serious failure that requires attention and prevents the
             device from transmitting data but that could be recovered
             without rebooting the system.";
        }
        enum error {
          value 4;
          description
            "Indicate that a failure occurred that could interrupt the normal data
             flow but that does not cause the device to re-register.";
        }
        enum warning {
          value 5;
          description
            "Indicate that a failure occurred that could interrupt the normal data
             flow but that does not cause the device to re-register.";
        }
        enum notice {
          value 6;
          description
            "Indicate a milestone or checkpoint in normal operation that could be of
             particular importance for troubleshooting.";
        }
        enum information {
```

```
          value 7;
          description
            "Indicate a milestone or checkpoint in normal operation that could be of
             particular importance for troubleshooting.";
        }
        enum debug {
          value 8;
          description
            "Reserved for vendor-specific events.";
        }
      }
      mandatory true;
      description
        "The priority level of this event, as defined by the vendor.  These are
         ordered from most serious (emergency) to least serious (debug)
         During normal operation, no event more critical than notice(6)
         should be generated.  Events between warning and emergency
         should be generated at appropriate levels of problems (e.g.
         emergency when the box is about to crash).";
      reference
        "DOCS-CABLE-DEVICE-MIB.docsDevEvLevel";
    }
    leaf event-id {
      type uint32;
      mandatory true;
      description
        "For this product, uniquely identifies the type of event that is reported
         by this entry.";
      reference
        "DOCS-CABLE-DEVICE-MIB.docsDevEvId";
    }
    leaf event-message {
      type string {
        length "0..255";
      }
      mandatory true;
      description
        "Provides a human-readable description of the event, including all
         relevant context (interface numbers, etc.).";
      reference
        "DOCS-CABLE-DEVICE-MIB.docsDevEvText";
    }
  }
}
```

Annex F (normative): Detailed MIB Requirements

F.1 Conventions Used in this Annex

The abbreviations and rules in table F.1 apply to this Annex.

Table F.1: MIB Implementation Support

Requirement Type	Table Notation	Description
Mandatory	M	The object shall be implemented correctly according to the MIB definition.
Not Applicable	NA	Not applicable to the device.
Optional	O	A vendor can choose to implement or not implement the object. If a vendor chooses to implement the object, the object shall be implemented correctly according to the MIB definition. If a vendor chooses not to implement the object, an agent shall respond with the appropriate error/exception condition for a non-existent object (e.g. "noSuchObject" for SNMPv2c).

F.2 CCAP-MIB Object Details

Table F.2 lists the CCAP compliance requirements summary.

Table F.2: CCAP-MIB Compliance Requirements

CCAP-MIB (Annex A.1)		
ccapInterfaceIndexMapTable		
Objects	**Requirement**	**Access**
ccapInterfaceIndexMapEntry	M	N-Acc
ccapInterfaceIndexMapPath	M	RO
ccapInterfaceIndexMapEntPhysicalIndex	M	RO
ccapMpegInputProgTable		
Objects	**Requirement**	**Access**
ccapMpegInputProgEntry	M	N-Acc
ccapMpegInputProgBitRate	M	RO
ccapMpegInputProgRequestedBandwidth	M	RO
ccapMpegOutputProgTable		
Objects	**Requirement**	**Access**
ccapMpegOutputProgEntry	M	N-Acc
ccapMpegOutputProgBitRate	M	RO
ccapMpegInputProgVideoSessionTable		
Objects	**Requirement**	**Access**
ccapMpegInputProgVideoSessionEntry	M	N-Acc
ccapMpegInputProgVideoSessionStatus	M	RO
ccapMpegOutputProgVideoSessionTable		
Objects	**Requirement**	**Access**
ccapMpegOutputProgVideoSessionEntry	M	N-Acc
ccapMpegOutputProgVideoSessionStatu s	M	RO
ccapEcmgStatusTable		
Objects	**Requirement**	**Access**
ccapEcmgStatusEntry	M	N-Acc
ccapEcmgIndex	M	N-Acc
ccapEcmgNumActiveSessions	M	RO
ccapEcmgCwMessageCount	M	RO

CCAP-MIB (Annex A.1)		
ccapEcmdStatusTable		
Objects	**Requirement**	**Access**
ccapEcmdStatusEntry	M	N-Acc
ccapEcmdIndex	M	N-Acc
ccapEcmdNumActiveSessions	M	RO
ccapEcmdCwMessageCount	M	RO
ccapMpegDecryptSessionTable		
Objects	**Requirement**	**Access**
ccapMpegDecryptSessionEntry	M	N-Acc
ccapMpegDecryptSessionDecrypted	M	RO

F.3 HMS-MIB Object Details

Table F.3 lists the CCAP compliance requirements summary.

Table F.3: CCAP HMS-MIB compliance requirements

SCTE-HMS-QAM-MIB [28]		
qamChannelTable		
Objects	**Requirement**	**Access**
qamChannelFrequency	M	RO
qamChannelModulationFormat	M	RO
qamChannelInterleaverLevel	M	RO
qamChannelInterleaverMode	M	RO
qamChannelPower	M	RO
qamChannelSquelch	M	RO
qamChannelContWaveMode	M	RO
qamChannelAnnexMode	M	RO
qamChannelCommonTable		
Objects	**Requirement**	**Access**
qamChannelCommonOutputBw	M	RO
qamChannelCommonUtilization	M	RO
qamConfigTable		
Objects	**Requirement**	**Access**
qamConfigIndex	M	N-Acc
qamConfigQamChannelIdMin	M	RO
qamConfigQamChannelIdMax	M	RO
qamConfigIPAddrType	M	RO
qamConfigIPAddr	M	RO
qamConfigUdpPortRangeMin	M	RO
qamConfigUdpPortRangeMax	M	RO
qamConfigOutputProgNoMin	M	RO
qamConfigOutputProgNoMax	M	RO
SCTE-HMS-MPEG-MIB [29]		
mpegDigitalInputs		
Object	**Requirement**	**Access**
mpegLossOfSignalTimeout	M	RO
mpegInputTSTable		
Objects	**Requirement**	**Access**
mpegInputTSIndex	M	N-Acc
mpegInputTSType	M	RO
mpegInputTSConnectionType	M	RO
mpegInputTSConnection	M	RO
mpegInputTSActiveConnection	M	RO
mpegInputTSPsiDetected	M	RO
mpegInputTSStartTime	M	RO
mpegInputTSResourceAllocated	M	RO
mpegInputTSNumPrograms	M	RO
mpegInputTSRate	M	RO

SCTE-HMS-QAM-MIB [28]		
mpegInputTSMaxRate	M	RO
mpegInputTSPatVersion	M	RO
mpegInputTSCatVersion	M	RO
mpegInputTSNitPid	M	RO
mpegInputTSNumEmms	M	RO
mpegInputTSTSID	M	RO
mpegInputTSLock	O	RO
mpegInputProgTable		
Objects	**Requirement**	**Access**
mpegInputProgIndex	M	N-Acc
mpegInputProgNo	M	RO
mpegInputProgPmtVersion	M	RO
mpegInputProgPmtPid	M	RO
mpegInputProgPcrPid	M	RO
mpegInputProgEcmPid	M	RO
mpegInputProgNumElems	M	RO
mpegInputProgNumEcms	M	RO
mpegInputProgCaDescr	M	RO
mpegInputProgScte35Descr	O	RO
mpegInputProgScte18Descr	O	RO
mpegProgESTable		
Objects	**Requirement**	**Access**
mpegProgESIndex	M	N-Acc
mpegProgESPID	M	RO
mpegProgESType	M	RO
mpegProgESCaDescr	M	RO
mpegProgESScte35Descr	O	RO
mpegProgESScte18Descr	O	RO
mpegInputStatsTable		
Objects	**Requirement**	**Access**
mpegInputStatsPcrJitter	M	RO
mpegInputStatsMaxPacketJitter	M	RO
mpegInputStatsPcrPackets	M	RO
mpegInputStatsNonPcrPackets	M	RO
mpegInputStatsUnexpectedPackets	M	RO
mpegInputStatsContinuityErrors	M	RO
mpegInputStatsSyncLossPackets	M	RO
mpegInputStatsPcrIntervalExceeds	M	RO
mpegInputUdpOriginationTable		
Objects	**Requirement**	**Access**
mpegInputUdpOriginationIndex	M	N-Acc
mpegInputUdpOriginationId	M	N-Acc
mpegInputUdpOriginationIfIndex	M	RO
mpegInputUdpOriginationInetAddrType	M	RO
mpegInputUdpOriginationSrcInetAddr	M	RO
mpegInputUdpOriginationDestInetAddr	M	RO
mpegInputUdpOriginationDestPort	M	RO
mpegInputUdpOriginationActive	M	RO
mpegInputUdpOriginationPacketsDetected	M	RO
mpegInputUdpOriginationRank	M	RO
mpegInputUdpOriginationInputTSIndex	M	RO
mpegInsertPacketTable		
Objects	**Requirement**	**Access**
mpegInsertPacketIndex	M	N-Acc
mpegInsertPacketListId	M	RO
mpegInsertPacketImmediateExecution	M	RO
mpegInsertPacketStartTime	M	RO
mpegInsertPacketRepeat	M	RO
mpegInsertPacketContinuousFlag	M	RO
mpegInsertPacketRate	M	RO
mpegInsertPacketDeviceIndex	M	RO
mpegOutputStatsTable		
Objects	**Requirement**	**Access**
mpegOutputStatsDroppedPackets	M	RO

SCTE-HMS-QAM-MIB [28]		
mpegOutputStatsFifoOverflow	M	RO
mpegOutputStatsFifoUnderflow	M	RO
mpegOutputStatsDataRate	M	RO
mpegOutputStatsAvailableBandwidth	M	RO
mpegOutputStatsChannelUtilization	M	RO
mpegOutputStatsTotalPackets	M	RO
mpegOutputTSTable		
Objects	**Requirement**	**Access**
mpegOutputTSIndex	M	N-Acc
mpegOutputTSType	M	RO
mpegOutputTSConnectionType	M	RO
mpegOutputTSConnection	M	RO
mpegOutputTSNumPrograms	M	RO
mpegOutputTSTSID	M	RO
mpegOutputTSNitPid	M	RO
mpegOutputTSCaPid	M	RO
mpegOutputTSCatInsertRate	M	RO
mpegOutputTSPatInsertRate	M	RO
mpegOutputTSPmtInsertRate	M	RO
mpegOutputTSStartTime	M	RO
mpegOutputProgTable		
Objects	**Requirement**	**Access**
mpegOutputProgIndex	M	N-Acc
mpegOutputProgNo	M	RO
mpegOutputProgPmtVersion	M	RO
mpegOutputProgPmtPid	M	RO
mpegOutputProgPcrPid	M	RO
mpegOutputProgEcmPid	M	RO
mpegOutputProgNumElems	M	RO
mpegOutputProgNumEcms	M	RO
mpegOutputProgCaDescr	M	RO
mpegOutputProgScte35Descr	O	RO
mpegOutputProgScte18Descr	O	RO
mpegOutputProgElemStatsTable		
Objects	**Requirement**	**Access**
mpegOutputProgElemStatsIndex	M	N-Acc
mpegOutputProgElemStatsPid	M	RO
mpegOutputProgElemStatsElemType	M	RO
mpegOutputProgElemStatsDataRate	O	RO
mpegOutputUdpDestinationTable		
Objects	**Requirement**	**Access**
mpegOutputUdpDestinationIndex	NA	
mpegOutputUdpDestinationId	NA	
mpegOutputUdpDestinationIfIndex	NA	
mpegOutputUdpDestinationInetAddrType	NA	
mpegOutputUdpDestinationSrcInetAddr	NA	
mpegOutputUdpDestinationDestInetAddr	NA	
mpegOutputUdpDestinationDestPort	NA	
mpegOutputUdpDestinationOutputTSIndex	NA	
mpegProgramMappingTable		
Objects	**Requirement**	**Access**
mpegProgramMappingIndex	M	N-Acc
mpegProgramMappingOutputProgIndex	M	RO
mpegProgramMappingOutputTSIndex	M	RO
mpegProgramMappingInputProgIndex	M	RO
mpegProgramMappingInputTSIndex	M	RO
mpegVideoSessionTable		
Objects	**Requirement**	**Access**
mpegVideoSessionIndex	M	N-Acc
mpegVideoSessionPhyMappingIndex	M	RO
mpegVideoSessionPIDRemap	M	RO
mpegVideoSessionMode	M	RO
mpegVideoSessionState	M	RO
mpegVideoSessionProvMethod	M	RO

SCTE-HMS-QAM-MIB [28]		
mpegVideoSessionEncryptionType	M	RO
mpegVideoSessionEncryptionInfo	M	RO
mpegVideoSessionBitRate	M	RO
mpegVideoSessionID	M	RO
mpegVideoSessionSelectedInput	M	RO
mpegVideoSessionSelectedOutput	M	RO
mpegVideoSessionPtrTable		
Objects	**Requirement**	**Access**
mpegVideoSessionPtrInputProgIndex	M	N-Acc
mpegVideoSessionPtrInputTSIndex	M	RO
mpegVideoSessionPtrInputTSConnType	M	RO
mpegVideoSessionPtrInputTSConnection	M	RO
mpegVideoSessionPtrOutputProgIndex	M	RO
mpegVideoSessionPtrOutputTSIndex	M	RO
mpegVideoSessionPtrOutputTSConnType	M	RO
mpegVideoSessionPtrOutputTSConnection	M	RO
mpegVideoSessionPtrStatus	M	RO
mpegInputTSOutputSessionTable		
Objects	**Requirement**	**Access**
mpegInputTSOutputSessionCreateTime	M	RO

Annex G (normative): YANG Configuration Module

```
module ccap {
    namespace "urn:cablelabs:params:xml:ns:yang:ccap";
    prefix ccap;
    import ietf-inet-types {
        prefix inet;
        revision-date 2010-09-24;
    }
    import ietf-yang-types {
        prefix yang;
        revision-date 2010-09-24;
    }
    organization "Cable Television Laboratories, Inc.";
    contact
        "Postal: Cable Television Laboratories, Inc.
         858 Coal Creek Circle
         Louisville, Colorado 80027-9750
         U.S.A.
         Phone: +1 303-661-9100
         Fax:   +1 303-661-9199
         E-mail: mibs@cablelabs.com";

    description "This module contains a collection of groupings and data definition statements
related to CCAP system configuration and state.";
    reference "Data-Over-Cable Service Interface Specifications CCAP Operations Support System
Interface Specification, CM-SP-CCAP-OSSI, Cable Television Laboratories, Inc.";
    revision 2013-08-08 {
        description "CCAP-OSSI I05 version.";
    }

    grouping host {
      choice address-or-hostname {
        mandatory true;
        case address {
          leaf address {
            mandatory true;
            type inet:ip-address;
          }
        }
        case name {
          leaf name {
            mandatory true;
            type inet:domain-name;
          }
        }
     }
     description
      "The host type represents either an strongly-typed IP address or a DNS
       domain name. Use of this type avoids the weak validation inherent in the
       union-based inet:host type as with this type an ip-address cannot be
       inappropriately validated as a domain-name accidentally.";
   }

    typedef ccap-octet-data-type {
        type string {
            pattern "([0-9a-fA-F]{2})*";
        }
        description "A derived type representing the lexical value space of XML Schema hexBinary
defined as 'each binary octet is encoded as a character tuple, consisting of two hexadecimal digits
([0-9a-fA-F]) representing the octet code.' Please note that length constraints on this derived type
needs to be in multiples of 2 to avoid conflicts between length and pattern space";
        reference "[XML-Schema] 3.2.15 hexBinary";
    }

  typedef ip-host-prefix {
    type union {
      type ipv4-host-prefix;
      type ipv6-host-prefix;
    }
    description
     "The ip-prefix type represents an IP host address plus prefix and is IP
      version neutral.  The format of the textual representations
```

```
    implies the IP version. This type is similar to inet:ip-prefix.";
}

typedef ipv4-host-prefix {
  type string {
    pattern
       '(([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.){3}'
     +  '([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])'
     + '/(([0-9])|([1-2][0-9])|(3[0-2]))';
  }
  description
   "The ipv4-host-prefix type represents an IPv4 host address
    plus the prefix length, separated by a slash.
    The prefix length is given by the number following the
    slash character and is less than or equal to 32.

    A prefix length value of n corresponds to an IP address
    mask that has n contiguous 1-bits from the most
    significant bit (MSB) and all other bits set to 0.

    This type is derived from the inet:ipv4-prefix type,
    which has all bits of the IPv4 address set to zero that
    are not part of the IPv4 prefix. Use of that type requires
    separate configuration of the interface host address.";
}

typedef ipv6-host-prefix {
  type string {
    pattern '((:|[0-9a-fA-F]{0,4}):)([0-9a-fA-F]{0,4}:){0,5}'
          + '((([0-9a-fA-F]{0,4}:)?(:|[0-9a-fA-F]{0,4}))|'
          + '(((25[0-5]|2[0-4][0-9]|[01]?[0-9]?[0-9])\.){3}'
          + '(25[0-5]|2[0-4][0-9]|[01]?[0-9]?[0-9])))'
          + '(/(([0-9])|([0-9]{2})|(1[0-1][0-9])|(12[0-8])))';
    pattern '(([^:]+:){6}(([^:]+:[^:]+)|(.*\..*))|'
          + '((([^:]+:)*[^:]+)?::(([^:]+:)*[^:]+)?)'
          + '(/.+)';
  }
  description
   "The ipv6-host-prefix type represents an IPv4 host address
    plus the prefix length, separated by a slash.
    The prefix length is given by the number following the
    slash character and isless than or equal to 128.

    A prefix length value of n corresponds to an IP address
    mask that has n contiguous 1-bits from the most
    significant bit (MSB) and all other bits set to 0.

    This type is derived from the inet:ipv6-prefix type,
    which has all bits of the IPv6 address set to zero that
    are not part of the IPv6 prefix. Use of that type requires
    separate configuration of the interface host address.

    The IPv6 address is represented
    in the compressed format described in RFC 4291 [i.16], Section
    2.2, item 2 with the following additional rules: the ::
    substitution is to be applied to the longest sequence of
    all-zero 16-bit chunks in an IPv6 address.  If there is
    a tie, the first sequence of all-zero 16-bit chunks is
    replaced by ::.  Single all-zero 16-bit chunks are not
    compressed.  The canonical format uses lowercase
    characters and leading zeros are not allowed.";
  reference
   "RFC 4291: IP Version 6 Addressing Architecture";
}
  typedef acl-comparator-type {
      type enumeration {
          enum other {
              value 0;
          }
          enum lt {
              value 1;
          }
          enum gt {
              value 2;
          }
          enum eq {
              value 3;
          }
```

```
            enum neq {
                value 4;
            }
        }
    }
    typedef load-bal-init-tech-type {
      type bits {
          bit reinit-mac {
              position 0;
          }
          bit broadcast-ranging {
              position 1;
          }
          bit unicast-ranging {
              position 2;
          }
          bit ranging {
              position 3;
          }
          bit direct {
              position 4;
          }
          bit disable {
              position 5;
          }
     }
      description "Initialization techniques used when changing channels in load balancing groups.
                   An empty value means no channel changes allowed.";
    }
    typedef load-balancing-rule-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum enabled {
                value 2;
            }
            enum disabled {
                value 3;
            }
            enum disabled-period {
                value 4;
            }
        }
    }
    typedef interleaver-depth-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum fecI8J16 {
                value 2;
            }
            enum fecI12J17 {
                value 3;
            }
            enum fecI16J8 {
                value 4;
            }
            enum fecI32J4 {
                value 5;
            }
            enum fecI64J2 {
                value 6;
            }
            enum fecI128J1 {
                value 7;
            }
            enum fecI128J2 {
                value 8;
            }
            enum fecI128J3 {
                value 9;
            }
```

```
            enum fecI128J4 {
                value 10;
            }
            enum fecI128J5 {
                value 11;
            }
            enum fecI128J6 {
                value 12;
            }
            enum fecI128J7 {
                value 13;
            }
            enum fecI128J8 {
                value 14;
            }
        }
    }
    typedef qam-modulation-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum qam64 {
                value 2;
            }
            enum qam128 {
                value 3;
            }
            enum qam256 {
                value 4;
            }
            enum qam512 {
                value 5;
            }
            enum qam1024 {
                value 6;
            }
        }
        description "This value defines the type of Downstream QAM Modulation.";
    }
    typedef ca-encryptor-type-type {
        type enumeration {
            enum other;
            enum motorola;
            enum cisco;
            enum simulcrypt;
        }
    }
    typedef encryption-scheme-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum des {
                value 2;
            }
            enum aes {
                value 3;
            }
            enum 3des {
                value 4;
            }
            enum dvbcsa {
                value 5;
            }
            enum dvbcsa3 {
                value 6;
            }
        }
    }
    typedef encryption-algorithm-ctl-type {
        type enumeration {
        enum other {
            value 1;
```

```
        description "The value of other is used when a vendor-extension has been implemented for
this attribute.";
        }
        enum cmts {
            value 2;
        }
        enum mgmt {
            value 3;
        }
    }
    description "This enumerates the set of ways a CMTS can select the encryption algorithm. Whether
the CMTS can select the algorithm or if this can be set manually using the Alg attribute. If this
attribute is set to 'cmts', the CMTS can select the encryption algorithm for the Security
Association (SA). If this attribute is set to 'mgmt', the Alg attribute is used to define the
encryption algorithm for this SA.";
    }
    typedef encryption-algorithm-type {
        type enumeration {
        enum other {
            value 1;
        description "The value of other is used when a vendor-extension has been implemented for
this attribute.";
        }
        enum des56-cbc-mode {
            value 2;
        }
        enum des40-cbc-mode {
            value 3;
        }
        enum aes128-cbc-mode {
            value 4;
        }
    }
    description "This enumerates the set of possible encryption algorithms which can be used for a
Security Association";
    }
    typedef cert-revocation-method-type {
      type enumeration {
      enum other {
          value 1;
          description "The value of other is used when a vendor-extension has been implemented for
this attribute.";
      }
      enum none {
          value 2;
      }
      enum crl {
          value 3;
      }
      enum ocsp {
          value 4;
      }
      enum crl-and-ocsp {
          value 5;
      }
      }
      description
    "This enumerates the set of certificate revocation methods which can used by the CMTS to verify
the cable modem certificate validity. The certificate revocation methods include Certification
Revocation List (CRL) and Online Certificate Status Protocol (OCSP).
         The following options are available:
             The option 'none' indicates that the CMTS does not attempt to determine the revocation
status of a certificate.
             The option 'crl' indicates the CMTS uses a Certificate Revocation List (CRL) as defined
by the Url attribute of the CmtsCertRevocationList object. When the value of this attribute is
changed to 'crl', it triggers the CMTS to retrieve the CRL file from the URL specified by the Url
attribute. If the value of this attribute is 'crl' when the CMTS starts up, it triggers the CMTS to
retrieve the CRL file from the URL specified by the Url attribute.
             The option 'ocsp' indicates the CMTS uses the Online Certificate Status Protocol (OCSP)
as defined by the Url attribute of the CmtsOnlineCertStatusProtocol object.
             The option 'crlAndOcsp' indicates the CMTS uses both the CRL as defined by the Url
attribute in the CmtsCertRevocationList object and OCSP as defined by the Url attribute in the
CmtsOnlineCertStatusProtocol object.";
    }
    typedef filter-action-type {
        type enumeration {
        enum other {
            value 1;
```

```
        description "The value of other is used when a vendor-extension has been implemented for
this attribute.";
        }
        enum permit {
            value 2;
        }
        enum deny {
            value 3;
        }
    }
    description "This enumerates the set of actions which can be taken upon this filter matching.
'permit' means to stop the classification matching and accept the packet for further processing.
'deny' means to drop the packet.";
    }
    typedef upstream-fec-mode-type {
        type enumeration {
            enum other;
            enum enabled;
            enum disabled;
            enum perOnu;
        }
        reference
            "CCAP Operations Support System Interface Specification
             CM-SP-CCAP-OSSI-I01-110930 UpstreamFecMode section.";
    }
    typedef downstream-fec-mode-type {
        type enumeration {
            enum other;
            enum enabled;
            enum disabled;
            enum perOnu;
        }
        reference
            "CCAP Operations Support System Interface Specification
             CM-SP-CCAP-OSSI-I01-110930 DownstreamFecMode section.";
    }
    typedef downstream-allocation-type {
        type enumeration {
        enum other {
            value 1;
          description "The value of other is used when a vendor-extension has been implemented for
this attribute.";
        }
        enum docsis-only {
            value 2;
        }
        enum video-only {
            value 3;
        }
        enum any {
            value 4;
        }
    }
    description "This enumerates the services a DownChannel instance can be allocated to. A value of
'any' means that the ERM could configure the QAM resource for either video or DOCSIS. The value of
other(1) is used when a vendor-extension has been implemented for this attribute.";
    }
    typedef cipher-type {
        type bits {
            bit other {
                position 0;
            }
            bit des3 {
                position 1;
                description "Really 3des, but that is not allowed in YANG.";
            }
            bit aes128 {
                position 2;
            }
            bit aes192 {
                position 3;
            }
            bit aes256 {
                position 4;
            }
            bit arcfour {
                position 5;
            }
```

```
        bit blowfish {
            position 6;
        }
        bit cast {
            position 7;
        }
        bit twofish128 {
            position 8;
        }
        bit twofish192 {
            position 9;
        }
        bit twofish256 {
            position 10;
        }
    }
    description "Set of encryption algorithms uses, e.g. for ssh.";
}
typedef auth-code-type {
    type bits {
        bit other {
            position 0;
        }
        bit md5 {
            position 1;
        }
        bit md5-96 {
            position 2;
        }
        bit sha1 {
            position 3;
        }
        bit sha1-96 {
            position 4;
        }
        bit ripemd-160 {
            position 5;
        }
        bit sha2-256 {
            position 6;
        }
        bit sha2-512 {
            position 7;
        }
    }
    description "Set of message authentication algorithms used, e.g. for ssh.";
}
typedef host-auth-type {
    type enumeration {
        enum other {
            value 0;
        }
        enum none {
            value 1;
        }
        enum ssh-dss {
            value 2;
        }
        enum ssh-rsa {
            value 3;
        }
        enum pgp-sign-rsa {
            value 4;
        }
        enum pgp-sign-dss {
            value 5;

        }
    }
    description "Public key format used, e.g. in ssh host authentication.";
}
typedef trigger-flag-type {
    type bits {
        bit registration {
            position 0;
        }
        bit ranging-retry {
            position 1;
```

```
            }
        }
        description "This data type defines the union of Diagnostic Log trigger types. Bit 0
represents the registration trigger, Bit 1 represents the ranging retry trigger.";
        reference
            "DOCSIS 3.0 Operations Support System Interface Specification
             CM-SP-OSSIv3.0-I15-110623 Diagnostic Log Annex.";
    }
    typedef admin-state-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum up {
                value 2;
            }
            enum down {
                value 3;
            }
            enum testing {
                value 4;
            }
        }
        description "This data type defines the admin state. The value of 'other' is used when a
vendor extension has been implemented for this attribute.";
        reference "RFC 2863";
    }
    typedef attribute-mask-type {
        type bits {
            bit bonded {
                position 0;
            }
            bit low-latency {
                position 1;
            }
            bit high-availability {
                position 2;
            }
            bit reserved-3 {
                position 3;
            }
            bit reserved-4 {
                position 4;
            }
            bit reserved-5 {
                position 5;
            }
            bit reserved-6 {
                position 6;
            }
            bit reserved-7 {
                position 7;
            }
            bit reserved-8 {
                position 8;
            }
            bit reserved-9 {
                position 9;
            }
            bit reserved-10 {
                position 10;
            }
            bit reserved-11 {
                position 11;
            }
            bit reserved-12 {
                position 12;
            }
            bit reserved-13 {
                position 13;
            }
            bit reserved-14 {
                position 14;
            }
            bit reserved-15 {
                position 15;
```

```
            }
            bit operator-16 {
                position 16;
            }
            bit operator-17 {
                position 17;
            }
            bit operator-18 {
                position 18;
            }
            bit operator-19 {
                position 19;
            }
            bit operator-20 {
                position 20;
            }
            bit operator-21 {
                position 21;
            }
            bit operator-22 {
                position 22;
            }
            bit operator-23 {
                position 23;
            }
            bit operator-24 {
                position 24;
            }
            bit operator-25 {
                position 25;
            }
            bit operator-26 {
                position 26;
            }
            bit operator-27 {
                position 27;
            }
            bit operator-28 {
                position 28;
            }
            bit operator-29 {
                position 29;
            }
            bit operator-30 {
                position 30;
            }
            bit operator-31 {
                position 31;
            }
        }
    }
    typedef downstream-phy-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum dvbc {
                value 2;
            }
            enum j83annexB {
                value 3;
            }
            enum j83annexC {
                value 4;
            }
        }
    }
    typedef msc-state-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum channel-enabled {
                value 2;
```

```
            }
            enum channel-disabled {
                value 3;
            }
            enum dormant {
                value 4;
            }
        }
    }
    typedef us-channel-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum tdma {
                value 2;
            }
            enum atdma {
                value 3;
            }
            enum scdma {
                value 4;
            }
            enum tdmaAndAtdma {
                value 5;
            }
        }
        description "Indicates the DOCSIS Upstream Channel Type. ";
    }
    typedef us-channel-width {
        type enumeration {
        enum other;
        enum 200000;
        enum 400000;
        enum 800000;
        enum 1600000;
        enum 3200000;
        enum 6400000;
        }
    description "The upstream channel width in Hertz.";
    }
    typedef direction-type {
        type enumeration {
        enum other {
            value 1;
        description "The value of other is used when a vendor-extension has been implemented for
this attribute.";
        }
        enum downstream {
            value 2;
        }
        enum upstream {
            value 3;
        }
    }
    description "This enumerates the set of potential Direction values for the QoS Parameter Set.";
    }
    typedef ds-resequencing-type {
        type enumeration {
        enum other {
            value 1;
        description "The value of other is used when a vendor-extension has been implemented for
this attribute.";
        }
        enum resequencing-dsid {
            value 2;
        }
        enum no-resequencing-dsid {
            value 3;
        }
    }
    description "This enumerates the set of potential DsResequencing values for the QoS Parameter
Set.";
    }
    typedef service-flow-scheduling-type {
        type enumeration {
```

```
        enum other {
            value 1;
        description "The value of other is used when a vendor-extension has been implemented for
this attribute.";
        }
        enum best-effort {
            value 2;
        }
        enum non-real-time-polling-service {
            value 3;
        }
        enum real-time-polling-service {
            value 4;
        }
        enum unsolicited-grant-service-with-ad {
            value 5;
        }
        enum unsolicited-grant-service {
            value 6;
        }
    }
    description "This enumerates the set of possible SchedulingType values for the QoS Parameter
Set.";
    }
    typedef ip-prov-mode-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum ipv4-only {
                value 2;
            }
            enum ipv6-only {
                value 3;
            }
            enum alternate {
                value 4;
            }
            enum dual-stack {
                value 5;
            }
        }
        default ipv6-only;
    }
    typedef duplex-type {
        type enumeration {
        enum other {
            value 1;
        description "The value of other is used when a vendor-extension has been implemented for
this attribute.";
        }
        enum full-duplex {
            value 2;
        }
        enum half-duplex {
            value 3;
        }
        enum auto-negotiation {
            value 4;
        }
    }
    description "This enumerates the Ethernet DuplexStates of an interface can be in. The value of
other(1) is used when a vendor-extension has been implemented for this attribute.";
    }
    typedef ethernet-speed-type {
        type enumeration {
        enum other {
            value 1;
        description "The value of other is used when a vendor-extension has been implemented for
this attribute.";
        }
        enum ten-mb-ethernet {
            value 2;
        }
        enum hundred-mb-ethernet {
            value 3;
```

```
        }
        enum one-gb-ethernet {
            value 4;
        }
        enum auto {
            value 5;
        }
    }
    description "This enumerates the set of possible ethernet interface speeds. The value of
other(1) is used when a vendor-extension has been implemented for this attribute.";
    }
    typedef ethernet-protocol-id-type {
        type enumeration {
        enum other {
            value 1;
        description "The value of other is used when a vendor-extension has been implemented for
this attribute.";
        }
        enum none {
            value 2;
        }
        enum ethertype {
            value 3;
        }
        enum dsap {
            value 4;
        }
        enum mac {
            value 5;
        }
        enum all {
            value 6;
        }
    }
    description
        "This enumerates the set of formats of the layer 3 protocol ID in the Ethernet frame. A
value of 'none' means that the rule does not use the layer 3 protocol type as a matching criteria. A
value of 'ethertype' means that the rule applies only to frames that contain an EtherType value.
Ethertype values are contained in packets using the DEC-Intel- Xerox (DIX) encapsulation or the
[i.6] Sub-Network Access Protocol (SNAP) encapsulation formats. A value of 'dsap' means that the
rule applies only to frames using the IEEE802.3 encapsulation format with a Destination Service
Access Point (DSAP) other than 0xAA (which is reserved for SNAP). A value of 'mac' means that the
rule applies only to MAC management messages for MAC management messages. A value of 'all' means
that the rule matches all Ethernet frame. If the Ethernet frame contains an 802.1P/Q Tag header
(i.e. EtherType 0x8100), this attribute applies to the embedded EtherType field within the 802.1p/Q
header.
             The value 'mac' is only used for passing UDCs to CMs during Registration. The CMTS
ignores filter rules that include the value of this attribute set to 'mac' for CMTS enforced
upstream and downstream subscriber management filter group rules.";
    reference "RFC 1042 Sub-Network Access Protocol (SNAP) encapsulation formats.";
    }
    typedef pon-symmetry-type {
        type enumeration {
        enum other {
            value 1;
        description "The value of other is used when a vendor-extension has been implemented for
this attribute.";
        }
        enum symmetric-10x10 {
            value 2;
        }
        enum asymmetric-10x1 {
            value 3;
        }
    }
    description "This enumerates the set of possible PON interface speeds allowing for asymmetrical
upstream and downstream speeds. The value of other(1) is used when a vendor-extension has been
implemented for this attribute.";
    }
    typedef qos-control-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum single-session {
                value 2;
```

```
            }
            enum aggregate-session {
                value 3;
            }
        }
        description "This attribute identifies how Group Classifier Rules (GCRs) and Group Service
Flows (GSFs) are instantiated when multiple sessions match the (S,G) criteria of this entry. If
'singleSession', the CMTS creates a unique GCR and a unique GSF for the session. If this object's
value is 'aggregateSession', all sessions matching this criterion are aggregated into the same
GSF.";
    }
    typedef upstream-frequency-range-type {
        type enumeration {
            enum other {
                value 1;
                    description "The value of other is used when a vendor-extension has been
implemented for this attribute.";
            }
            enum standard {
                value 2;
            }
            enum extended {
                value 3;
            }
        }
        description "This attribute indicates in MDD messages the upstream frequency upper band edge
of an upstream Channel. A value 'standard' means Standard Frequency Range and a value 'extended'
means Extended Frequency Range.";
        reference "DOCSIS 3.0 Operations Support System Interface Specification CM-SP-OSSIv3.0-I15-
110623 MdCfg Object section.";
    }
    typedef early-auth-encrypt-control-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum disable-eae {
                value 2;
            }
            enum enable-eae-ranging-based-enforcement {
                value 3;
            }
            enum enable-eae-capability-based-enforcement {
                value 4;
            }
            enum enable-eae-total-enforcement {
                value 5;
            }
        }
        description "This attribute enables or disables early authentication and encryption (EAE)
signaling for the MAC Domain. It also defines the type of EAE enforcement in the case that EAE is
enabled. If set to 'disable-eae', EAE is disabled for the MAC Domain. If set to 'enable-eae-ranging-
based-enforcement', 'enable-eae-capability-based-enforcement' or 'enable-eae-total-enforcement', EAE
is enabled for the MAC Domain. The following EAE enforcement methods are defined in the case where
EAE signaling is enabled: - The option 'enable-eae-ranging-based-enforcement' indicates EAE is
enforced on CMs that perform ranging with a B-INIT-RNG-REQ message. - The option 'enable-eae-
capability-based-enforcement' indicates EAE is enforced on CMs that perform ranging with a B-INIT-
RNG-REQ message in which the EAE capability flag is set. The option 'enable-eae-total-enforcement'
indicates EAE is enforced on all CMs regardless of their EAE capabilities.";
        reference "DOCSIS 3.0 Operations Support System Interface SpecificationCM-SP-OSSIv3.0-I15-
110623 MdCfg Object section.";
    }
    typedef bpi2-enforce-control-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum disable {
                value 2;
            }
            enum qosCfgFileWithBpi2AndCapabPrivSupportEnabled {
                value 3;
            }
            enum qosCfgFileWithBpi2Enabled {
```

```
                value 4;
            }
            enum qosCfgFile {
                value 5;
            }
            enum total {
                value 6;
            }
        }
        description "This attribute indicates the level of BPI+ enforcement policies with the MAC
Domain.";
        reference "DOCSIS 3.0 Operations Support System Interface SpecificationCM-SP-OSSIv3.0-I15-
110623 MdCfg Object section.";
    }
    typedef enable-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum enable {
                value 2;
            }
            enum disable {
                value 3;
            }
        }
       description "This attribute enables the enforcement of Multicast Authorization feature. When
this attribute is set to 'enable', Multicast Authorization is enforced; otherwise, clients are
permitted to join any IP multicast session. The factory default value of this attribute is
'disable'.";
    }
    typedef authorization-action-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum accept {
                value 2;
            }
            enum deny {
                value 3;
            }
        }
        description "This attribute specifies the authorization action for a session join attempt.
The value 'accept' indicates that a multicast join request is allowed. The value 'deny' indicates
that a multicast join request is denied. This type is used for both default behavior (i.e. a session
'join' request matches no session rules) and explicitly configured behavior (i.e. a session 'join'
request matches the session rule)";
    }
    typedef modulation-interval-usage-code-type {
        type enumeration {
            enum request {
                value 1;
            }
            enum requestData {
                value 2;
            }
            enum initialRanging {
                value 3;
            }
            enum periodicRanging {
                value 4;
            }
            enum shortData {
                value 5;
            }
            enum longData {
                value 6;
            }
            enum advPhyShortData {
                value 9;
            }
            enum advPhyLongData {
                value 10;
```

```
            }
            enum ugs {
                value 11;
            }
        }
        description "An index into the Channel Modulation table that, when grouped with other
Interval Usage Codes, fully instantiates all modulation sets for a given upstream channel.";
    }
    typedef modulation-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum qpsk {
                value 2;
            }
            enum qam8 {
                value 3;
            }
            enum qam16 {
                value 4;
            }
            enum qam32 {
                value 5;
            }
            enum qam64 {
                value 6;
            }
            enum qam128 {
                value 7;
            }
        }
        description "The modulation type in transmissions using this IUC.";
    }
    typedef preamble-length-type {
        type uint16 {
            range "0..1536";
        }
        units bits;
        description "The preamble length for this modulation profile in bits.";
    }
    typedef fec-error-correction-type {
        type uint8 {
            range "0..16";
        }
        units Bytes;
        description "The number of correctable errored bytes (t) used in forward error correction
code.  The value of 0 indicates that no correction is employed.  The number of check bytes appended
will be twice this value.";
    }
    typedef fec-codeword-length-type {
        type uint8 {
            range "1..255";
        }
        units Bytes;
        description "The number of correctable errored bytes (t) used in forward error correction
code.";
    }
    typedef scrambler-seed-type {
        type uint16 {
            range "0..32767";
        }
        description "The 15-bit seed value for the scrambler polynomial.";
    }
    typedef preamble-type {
       type enumeration {
           enum other {
               value 1;
               description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum qpsk0 {
                value 2;
            }
            enum qpsk1 {
                value 3;
```

```
            }
        }
        description "Preamble type for DOCSIS 2.0 bursts.  ";
    }
    typedef scdma-interleaver-step-size-type {
        type uint8 {
            range "0 | 1..32";
        }
        description "S-CDMA Interleaver step size.";
    }
    typedef scdma-subframe-codes-type {
        type uint8 {
            range "0 | 1..128";
        }
        description "S-CDMA sub-frame size.";
    }
    typedef dsg-client-id-class-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum broadcast {
                value 2;
            }
            enum mac-address {
                value 3;
            }
            enum ca-system-id {
                value 4;
            }
            enum application-id {
                value 5;
            }
        }
        description "The DSG Client Identification type. A DSG client id of type broadcast(1)
received by all DSG client(s). A DSG client id of type macAddress(2) is received by the DSG client
that has been assigned with this MAC address where the first 3 bytes is the Organization Unique
Identifier (OUI). A DSG client id of type caSystemId(3) is received by the DSG client that has been
assigned a CA_system_ID. A DSG client ID of type applicationId(4) is received by the DSG client that
has been assigned an application ID. The value of other(1) is used when a vendor-extension has been
implemented for this attribute.";
    }
    typedef non-zero-seconds-type {
        type uint8 {
            range "1..60";
        }
        units seconds;
        description "The non-zero seconds in a minute.";
    }
    typedef ack-sequence-interval-type {
        type uint16 {
            range "1..500";
        }
        units records;
        description "The maximum number of unacknowledged records that can be sent by the CCAP IPDR
exporter before receiving an acknowledgement.";
    }
    typedef seconds-day-max-type {
        type uint32 {
            range "0..86400";
        }
        units seconds;
        description "The number of seconds between 0 and 1 day.";
    }
    typedef ipdr-streaming-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum time-interval {
                value 2;
            }
            enum ad-hoc {
                value 3;
```

```
            }
            enum event {
                value 4;
            }
            enum time-event {
                value 5;
            }
        }
        description "This attribute configures the type of IPDR streaming used for the session. See
the IPDR Service Definition Schemas section of [OSSIv3.0] for the streaming types supported by each
Service Definition. The value of other(1) is used when a vendor-extension has been implemented for
this attribute.";
    }
    typedef ipdr-service-definition-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum cmts-cm-service-flow-type {
                value 2;
            }
            enum cmts-cm-reg-status-type {
                value 3;
            }
            enum cmts-cm-us-stats-type {
                value 4;
            }
            enum cmts-ds-util-stats-type {
                value 5;
            }
            enum cmts-us-util-stats-type {
                value 6;
            }
            enum cmts-topology-type {
                value 7;
            }
            enum cpe-type {
                value 8;
            }
            enum diag-log-type {
                value 9;
            }
            enum diag-log-detail-type {
                value 10;
            }
            enum diag-log-event-type {
                value 11;
            }
            enum samis-type-1 {
                value 12;
            }
            enum samis-type-2 {
                value 13;
            }
            enum spectrum-measurement-type {
                value 14;
            }
        }
        description "This attribute configures the service type definition for this IPDR session.
See the IPDR Service Definition Schemas section of [OSSIv3.0] for the definitions and schemas of the
types defined in this enumeration. The value of other(1) is used when a vendor-extension has been
implemented for this attribute.";
    }
    typedef table-interval-type {
        type uint8 {
            range "0..32";
        }
        units "tables per second";
        description "A table interval expressed in tables/second.";
    }
    typedef video-pid-usage-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
```

```
            }
            enum emm {
                value 2;
            }
            enum nit {
                value 3;
            }
            enum cat {
                value 4;
            }
            enum pat {
                value 5;
            }
            enum fixed {
                value 6;
            }
            enum eas {
                value 7;
            }
            enum dsm-cc {
                value 8;
            }
            enum eiss {
                value 9;
            }
            enum etvbif {
                value 10;
            }
            enum video {
                value 11;
            }
            enum audio {
                value 12;
            }
        }
        description "This enumeration defines the type of the identified PID stream. This value is
used to understand what anchor table (i.e. PAT, CAT) would need to be updated, in case
PidRemapEnable is set to True and a remap is required. In case of type 'eas', the table sections of
the PID stream may need to be interleaved with other table sections that would be present on the
same OutputPid. 'dsm-cc' is used for digital storage media command and control. 'eiss' is used for
ETV Integrated Signaling Streams (Stream type 0xC0 or 0x05 w-descriptor tag 0xA2). 'etvbif' is used
for ETV Binary Interchange Format (Stream type 0xC0 or 0x05 w-descriptor tag 0xA1 OR Stream Type
0X0B). 'video' is used for MPEG2 video streams. 'audio' is used for MPEG2 audio streams. The value
of other(1) is used when a vendor-extension has been implemented for this attribute.";
    }
    typedef cci-level-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum copy-freely {
                value 2;
            }
            enum copy-one-generation {
                value 3;
            }
            enum copy-no-more {
                value 4;
            }
            enum copy-never {
                value 5;
            }
        }
        description "This attribute represents the Copy Control Indicator/Digital Rights protection
applicable to the program. It is forwarded to all active ECMGs to be encapsulated into ECMs. The
value of other(1) is used when a vendor-extension has been implemented for this attribute.";
    }
    typedef cit-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum clear {
                value 2;
```

```
            }
            enum set {
                value 3;
            }
        }
        description "This attribute represents the Constrained Image Trigger flag applicable to the
program. It is forwarded to all active ECMGs to be encapsulated into ECMs. The value of other(1) is
used when a vendor-extension has been implemented for this attribute.";
    }
    typedef rct-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum not-asserted {
                value 2;
            }
            enum required {
                value 3;
            }
        }
        description "This attribute represents the Redistribution Control Trigger flag applicable to
the program. It is forwarded to all active ECMGs to be encapsulated into ECMs. The value of other(1)
is used when a vendor-extension has been implemented for this attribute.";

    }
    typedef cci-reserved-type {
        type uint8 {
            range "0..3";
        }
        description "This attribute reserves 2 bits of copy control information (CCI) for future
use. It is forwarded to all active ECMGs to be encapsulated into ECMs.";
    }
    typedef key-length-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum 56bits {
                value 2;
            }
            enum 128bits {
                value 3;
            }
            enum 192bits {
                value 4;
            }
            enum 256bits {
                value 5;
            }
        }
        description "This attribute configures the number of bits in the encryption keys used by
encryption algorithm defined by the EncryptionScheme attribute. The value of other(1) is used when a
vendor-extension has been implemented for this attribute.";
    }
    typedef erm-connection-mode-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum client {
                value 2;
            }
            enum server {
                value 3;
            }
            enum client-and-server {
                value 4;
            }
        }
        description "This attribute represents the type of TCP connection that is established by the
CCAP. The value can be one of the following:
```

```
- 'other' indicates that a vendor-extension has been implemented for this attribute. - 'client'
indicates that the CCAP has to initiate the TCP connection with the ERM. - 'server' indicates that
the CCAP has to wait the TCP connection from the ERM. - 'client-and-server' indicates that both the
CCAP or either the CCAP or the ERM can initiate the TCP connection.";
    }
    typedef integrated-server-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum ftp {
                value 2;
            }
            enum http {
                value 3;
            }
            enum ssh {
                value 4;
            }
            enum telnet {
                value 5;
            }
            enum netconf {
                value 6;
            }
        }
        description "The type of server being configured on the CCAP. The value of other(1) is used
when a vendor-extension has been implemented for this attribute. The CCAP should support a NETCONF
server type. other(1), ftp(2), http(3), ssh(4), telnet(5), netconf(6)";
    }
    typedef policy-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum login {
                value 2;
            }
            enum privileged-mode {
                value 3;
            }
        }
        description "This attribute is the first part of the key and configures the policy type for
the specified protocol.";
    }
    typedef protocol-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum radius {
                value 2;
            }
            enum tacacs-plus {
                value 3;
            }
            enum local-authentication {
                value 4;
            }
            enum none {
                value 5;
            }
        }
        description "This attribute is the second part of the key and represents the protocol used
by the AAA server. The value of other(1) is used when a vendor-extension has been implemented for
this attribute. ";
    }
    typedef timer-interval-type {
        type uint32 {
            range "5..2147483647";
        }
        units seconds;
```

```
        description "RIPv2 intervals.";
    }
    typedef send-version-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum rip-version-1 {
                value 2;
            }
            enum rip-version-2 {
                value 3;
            }
            enum rip-version-1-and-2 {
                value 4;
            }
        }
        description "This attribute configures the version of the RIP protocol being used for this
prefix. ";
    }
    typedef rip-auth-mode-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum no-authentication {
                value 2;
            }
            enum simple-password {
                value 3;
            }
            enum md5 {
                value 4;
            }
        }
        description "This attribute configures the mode for authentication. [RFC 2453] defines this
as simple password (type 2). The value of other(1) is used when a vendor-extension has been
implemented for this attribute.";
        reference "RFC 2453";
    }
    typedef application-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum host {
                value 2;
            }
            enum mta {
                value 3;
            }
            enum stb {
                value 4;
            }
            enum cm {
                value 5;
            }
            enum all {
                value 6;
            }
        }
        description "The device class for which this cable helper configuration applies.";
    }
    typedef igmp-version-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum igmp-v1 {
                value 2;
```

```
            }
            enum igmp-v2-or-mld-v1 {
                value 3;
            }
            enum igmp-v3-or-mld-v2 {
                value 4;
            }
        }
        description "The version of MGMD. Value 2 applies to IGMPv1 routers only. Value 3 applies to
IGMPv2 and MLDv1 routers, and value 4 applies to IGMPv3 and MLDv2 routers.";
    }
    typedef event-throttle-admin-state-type {
        type enumeration {
            enum unconstrained {
                value 1;
            }
            enum maintain-below-threshold {
                value 2;
            }
            enum stop-at-threshold {
                value 3;
            }
            enum inhibited {
                value 4;
            }
        }
        description "Refer to RFC 4639.";
        reference "RFC 4639";
    }
    typedef syslog-priority-type {
        type enumeration {
            enum emergency {
                value 1;
            }
            enum alert {
                value 2;
            }
            enum critical {
                value 3;
            }
            enum error {
                value 4;
            }
            enum warning {
                value 5;
            }
            enum notice {
                value 6;
            }
            enum information {
                value 7;
            }
            enum debug {
                value 8;
            }
        }
        description "See RFC 4639";
        reference "RFC 4639 docsDevEvPriority";
    }
    typedef syslog-reporting-type {
        type bits {
            bit local {
                position 0;
            }
            bit traps {
                position 1;
            }
            bit syslog {
                position 2;
            }
            bit local-volatile {
                position 3;
            }
            bit std-interface {
                position 4;
            }
        }
        description "See RFC 4639. The bit ordering has been changed from RFC 4639 to avoid gaps.";
```

```
        reference "RFC 4639 docsDevEvReporting";
    }
    typedef reg-detail-type {
        type bits {
            bit other {
                position 0;
            }
            bit initial-ranging {
                position 1;
            }
            bit ranging-auto-adj-complete {
                position 2;
            }
            bit start-eae {
                position 3;
            }
            bit start-dhcpv4 {
                position 4;
            }
            bit start-dhcpv6 {
                position 5;
            }
            bit dhcpv4-complete {
                position 6;
            }
            bit dhcpv6-complete {
                position 7;
            }
            bit start-config-file-download {
                position 8;
            }
            bit config-file-download-complete {
                position 9;
            }
            bit start-registration {
                position 10;
            }
            bit registration-complete {
                position 11;
            }
            bit bpi-init {
                position 12;
            }
            bit operational {
                position 13;
            }
        }
        description "Setting a bit representing a CM registration state will enable counting the
number of times the CMTS determines that such CM reaches that state as the last state before failing
to proceed further in the registration process and within the time interval considered for the CM
registration trigger detection.";
    }
    typedef ranging-retry-trigger-type {
        type enumeration {
            enum consecutive-miss {
                value 1;
            }
            enum miss-ratio {
                value 2;
            }
        }
        description "This attribute selects the type of ranging retry trigger to be enable in the
Diagnostic Log. A CM failure to perform ranging when a ranging opportunity is scheduled by the CMTS
is counted as ranging miss. Ranging retry trigger can be configured to either look at consecutive
ranging misses or ranging miss ratio over total number of station maintenance opportunities for a
certain time period. Setting this object to 'consecutiveMiss' will select consecutive ranging misses
as ranging retry trigger criteria. Setting this object to 'missRatio' will select ranging miss ratio
as ranging retry criteria.";

    }
    typedef notif-ctrl-type {
        type bits {
            bit high-threshold-reached {
                position 0;
            }
            bit low-threshold-reached {
                position 1;
            }
```

```
            bit full {
                position 2;
            }
        }
        description "This attribute is used to enable diagnostic log related notifications. Setting
bit 0 enables notification for reaching log size high threshold. Setting bit 1 enables notification
for returning back to log size low threshold after reaching log size high threshold. Setting bit 2
enables notification for Diagnostic Log size full.";
    }
    typedef snmp-access-type {
        type enumeration {
            enum read-only {
                value 1;
            }
            enum read-write {
                value 2;
            }
        }
        description "Defines the type of access granted to the SNMP request.";
    }
    typedef subtree-view-inclusion-type {
        type enumeration {
            enum other {
                value 1;
                description "The value of other is used when a vendor-extension has been implemented
for this attribute.";
            }
            enum included {
                value 2;
            }
            enum excluded {
                value 3;
            }
        }
        description "Indicates inclusion or exclusion of the subtree for the defined view.";
    }
    typedef notification-type {
        type enumeration {
            enum snmpv1-trap {
                value 1;
            }
            enum snmpv2c-trap {
                value 2;
            }
            enum snmpv2c-inform {
                value 3;
            }
        }
        description "Indicates the type of SNMP notification being sent. - 'snmpv1-trap': SNMP v1
trap - 'snmpv2c-trap': SNMP v2c trap - 'snmpv2c-inform': SNMP v2c Inform";
    }
    typedef address-prefix-len-type {
        type uint16 {
            range "0..2040";
        }
        description "This is based on RFC 4001 InetAddressPrefixLength.";
    }
    typedef snmp-tag-list-type {
        type string {
            length "0..255";
        }
        description "This data type is equivalent to the SnmpTagList.";
    }
    extension extensionPoint {
        description "Hint to pyang to, optionally, define this as being
                     of ext-type and to include ext-type in generated schema";
    }
    extension inlineType {
        description "Hint to pyang to, optionally, not to create a named complex
                     type, but rather leave this container or list as an in-line type.";
    }
    grouping vendor-extension-version {
        container vendor-extension-version
        {
            leaf major-version {
                type uint32;
                mandatory true;
```

```
                description "Major version provides the macro versioning number for each interface.
Versions containing the same major version should provide backwards compatibility.";
            }
            leaf minor-version {
                type int32;
                mandatory true;
                description "MinorVersion identifies incremental and backwards compatible updates to
a major version.";
            }
            leaf micro-version {
                type int32;
                description "MicroVersion is usually for bug fixes, without changes in
functionality.";
            }
            container yang-ext {
                ccap:extensionPoint; //different pyang flags impact use of this hint
                description "node for vendor YANG extensions";
            }
        }
    }
    grouping line-card-group {
        description "The line-card container contains configuration elements for a CCAP line card.
There are several types of line cards defined for the CCAP: Downstream (DLC), Upstream (ULC), System
Route Engine (SRE), a combined Upstream and Downstream line card, and an EPON line card.";
        reference
            "CCAP Operations Support System Interface Specification
             CM-SP-CCAP-OSSI-I01-110930 LineCard section.";
        leaf line-card-name {
            type string;
            mandatory true;
            description "This attribute stores the name of the line card being configured.  ";
        }
        leaf admin-state {
            type admin-state-type;
            default down;
            description "This attribute configures the administrative state of the line card.";
        }
        leaf protected-by {
            type leafref {
                path "../../../slot/slot-number";
            }
            description "Line card redundancy or sparing is achieved with a protect relationship
between two line cards.";
        }
    }
    grouping upstream-physical-channel-reference {
        description "A reference to an upstream physical channel.";
        leaf slot {
            type leafref {
                path "/ccap/chassis/slot/slot-number";
            }
            description "A reference to a slot number.";
        }
        leaf us-rf-port {
            type leafref {
                path "/ccap/chassis/slot[slot-number=current()/../slot]/us-rf-port/port-number";
            }
            description "A reference to a upstream RF port number.";
        }
        leaf upstream-physical-channel {
            type leafref {
                path "/ccap/chassis/slot[slot-number=current()/../slot]/us-rf-port[port-
number=current()/../us-rf-port]/upstream-physical-channel/channel-index";
            }
            description "A reference to an upstream physical channel.";
        }
    }
    grouping upstream-logical-channel-reference {
        description "A reference to an upstream logical channel.";
        leaf slot {
            type leafref {
                path "/ccap/chassis/slot/slot-number";
            }
            description "A reference to a slot number.";
        }
        leaf us-rf-port {
            type leafref {
                path "/ccap/chassis/slot[slot-number=current()/../slot]/us-rf-port/port-number";
```

```
            }
            description "A reference to a upstream RF port number.";
        }
        leaf upstream-physical-channel {
            type leafref {
                path "/ccap/chassis/slot[slot-number=current()/../slot]/us-rf-port[port-
number=current()/../us-rf-port]/upstream-physical-channel/channel-index";
            }
            description "A reference to an upstream physical channel.";
        }
        leaf upstream-logical-channel {
            type leafref {
                path "/ccap/chassis/slot[slot-number=current()/../slot]/us-rf-port[port-
number=current()/../us-rf-port]/upstream-physical-channel[channel-index=current()/../upstream-
physical-channel]/upstream-logical-channel/upstream-logical-channel-index";
            }
            description "A reference to an upstream logical channel.";
        }
    }
    grouping docsis-down-channel-reference {
        leaf slot {
            type leafref {
                path "/ccap/chassis/slot/slot-number";
            }
            description "A reference to a slot number.";
        }
        leaf ds-rf-port {
            type leafref {
                path "/ccap/chassis/slot[slot-number=current()/../slot]/ds-rf-port/port-number";
            }
            description "A reference to a downstream RF port number.";
        }
        leaf down-channel {
            type leafref {
                path "/ccap/chassis/slot[slot-number=current()/../slot]/ds-rf-port[port-
number=current()/../ds-rf-port]/down-channel/channel-index";
            }
            description "A reference to a DOCSIS downstream channel index.";
        }
    }
    grouping video-down-channel-reference {
        leaf slot {
            type leafref {
                path "/ccap/chassis/slot/slot-number";
            }
            description "A reference to a slot number.";
        }
        leaf ds-rf-port {
            type leafref {
                path "/ccap/chassis/slot[slot-number=current()/../slot]/ds-rf-port/port-number";
            }
            description "A reference to a downstream RF port number.";
        }
        leaf down-channel {
            type leafref {
                path "/ccap/chassis/slot[slot-number=current()/../slot]/ds-rf-port[port-
number=current()/../ds-rf-port]/down-channel/channel-index";
            }
            description "A reference to a video downstream channel index.";
        }
    }
    grouping ds-rf-port-reference {
        leaf slot {
            type leafref {
                path "/ccap/chassis/slot/slot-number";
            }
            description "Reference to the slot in which the downstream RF port resides.";
        }
        leaf ds-rf-port {
            type leafref {
                path "/ccap/chassis/slot[slot-number=current()/../slot]/ds-rf-port/port-number";
            }
            description "A reference to a downstream RF port number.";
        }
    }
    grouping us-rf-port-reference {
        leaf slot {
            type leafref {
```

```
                path "/ccap/chassis/slot/slot-number";
            }
            description "Reference to the slot in which the upstream RF port resides.";
        }
        leaf us-rf-port {
            type leafref {
                path "/ccap/chassis/slot[slot-number=current()/../slot]/us-rf-port/port-number";
            }
            description "A reference to an upstream RF port number.";
        }
    }
    grouping port-group {
        leaf port-number {
            type uint32;
            description "The port-number attribute of Port is a zero- or one-based index that
sequentially numbers the physical ports of each derived type. For example, the Port numbers of
'DsRfPort' objects start at zero and increase to n-1.";
        }
        leaf admin-state {
            type admin-state-type;
            default down;
            description "This attribute configures the administrative state of the physical port.";
        }
        leaf up-down-trap-enabled {
            type boolean;
            default false;
            description "Indicates whether linkUp/linkDown SNMP traps should be generated for this
interface.";
        }
    }
    grouping encryptor-group {
        leaf encryptor-index {
            type uint32;
            description "This key attribute identifies the instance of the encryptor.";
        }
        leaf ca-encryptor-type {
            type ca-encryptor-type-type;
            mandatory true;
            description "This enumeration defines the type of CA encryption the Encryptor uses. ";
        }
        leaf ecm-timeout {
            type uint16;
            units seconds;
            default 10;
            description "This attribute configures the number of seconds that a CCAP will wait to
get a response from a ECMG before switching to the redundant unit.";
        }
        leaf clear-stream-timeout {
            type uint16;
            units seconds;
            default 10;
            description "This configured attribute defines the number of seconds a given stream may
be sent in the clear when the stream is configured to be encrypted.  If this timer expires and the
session has not received any encryption information from the ECMG, the CCAP shall discontinue
forwarding this stream.";
        }
        list ecmg-usage {
            key ecmg-usage-index;
            min-elements 1;
            description "The ecmg-usage object provides for the configuration of multiple encryption
sessions. It is an intermediate object that provides linkages between Encryptor objects and the
ECMG(s) associated with those encrypted streams.";
            leaf ecmg-usage-index {
                type uint32;
                description "This is an index for an instance of this object. ecmg-usage-index is a
pointer to an active ECMG that can be used for any program session that requires encryption as long
as the CAS identifier matches.";
            }
            leaf priority {
                type uint32;
                mandatory true;
                description "This is the configured selection priority for any program session that
requires encryption when multiple ECMGs with the same CAS identifier are active. The ECMG with the
lowest number should be selected first.";
            }
            leaf ecmg-ref {
                type leafref {
                    path "/ccap/video/ecmg/ecm-index";
```

```
                }
                description "A reference to the index of an ecmg instance.";
            }
            container yang-ext {
                ccap:extensionPoint; //different pyang flags impact use of this hint
                description "node for vendor YANG extensions";
            }
        }
    }
    grouping mac-domain-configuration-group {
        description "The mac-domain-configuration-group configures the MAC domain attributes used by
DOCSIS and EPON MAC domains.";
        leaf mac-domain-name {
            type string;
            description "The name of the MacDomain.";
        }
        leaf ip-provisioning-mode {
            type ip-prov-mode-type;
            mandatory true;
            description
                "This attribute configures the IP provisioning mode for a MAC Domain.
                 When this attribute is set to 'ipv4-only' the CM will acquire a single IPv4 address
for the CM management stack.
                 When this attribute is set to 'ipv6-only' the CM will acquire a single IPv6 address
for the CM management stack.
                 When this attribute is set to 'alternate' the CM will acquire a single IPv6 address
for the CM management stack and, if failures occur, the CM will fall back to provisioning and
operation with an IPv4 address.
                 When this attribute is set to 'dual-stack' the CM will acquire both an IPv6 and
IPv4 address for provisioning and operation.";
            reference
                "DOCSIS 3.0 Operations Support System Interface Specification
                 CM-SP-OSSIv3.0-I15-110623 MdCfg Object Section.";
        }
        leaf admin-state {
            type admin-state-type;
            default down;
            description "This attribute configures the administrative state of the mac-domain.";
        }
        leaf up-down-trap-enabled {
            type boolean;
            default true;
            description "Indicates whether linkUp/linkDown SNMP traps should be generated for this
interface.";
        }
    }
    grouping down-channel-group {
        leaf channel-index {
            type uint32;
            description "This key identifies a channel on a specific RF Port.";
        }
        leaf admin-state {
            type admin-state-type;
            default down;
            description "This attribute represents the administrative status of the channel. For
downstream channels, Setting the value to down(3) results in the channel being muted, and the value
of testing(4) is used to generate a continuous test wave on this QAM channel.";
        }
        leaf up-down-trap-enabled {
            type boolean;
            default true;
            description "Indicates whether linkUp/linkDown SNMP traps should be generated for this
interface.";
        }
        leaf power-adjust {
            type int16;
            units tenthdb;
            default 0;
            description "This attribute represents the power gain for the channel. It is expressed
in TenthdB.";
        }
        leaf frequency {
            type uint32;
            units hertz;
            mandatory true;
            description "This attribute specifies the center frequency of the channel. It is
expressed in Hertz. The CCAP shall not allow the frequency of any DownChannel instances to overlap
on a given downstream RF port.";
```

```
        }
        leaf rf-mute {
            type boolean;
            default false;
            description "This attribute configures the mute state for the specific DownChannel. If
set to true, the ifOperStatus of the VideoDownChannel or DocsisDownChannel associated with this
instance of DownChannel is set to 'down'. If set to false, no muting takes place. Operation while
muted is described in [DRFI]";
            reference "DOCSIS Downstream RF Interface Specification CM-SP-DRFI-I11-110210.";
        }
        leaf qam-alias {
            type string;
            description "This attribute represents the name of the QAM channel and is equivalent to
the ifAlias object in the if-MIB.";
        }
        leaf errp-advertising {
            type boolean;
            default true;
            description "This attribute represents the ERRP advertisement state of the QAM channel.
If set to true, the QAM channel is advertised; otherwise it is not advertised. This is primarily
useful when statically configuring the QAM channels and when the QAM channel is not made part of the
ERM channel list. This attribute is optional for DocsisDownChannel.";
        }
        choice erm-channel-config {
            case erm-managed {
                    description "This optional configuration object allows for the configuration of
the needed parameters that are communicated to an ERM for a given DownChannel object instance.";
                    leaf input-map-group-name {
                        type string;
                        mandatory true;
                        description "This attribute represents the address field in the
WithdrawnRoute and ReachableRoutes ERRP attributes. This attribute is optional for
DocsisDownChannel.";
                    }
                    leaf phy-lock-parameters {
                        type bits {
                            bit frequency {
                                position 0;
                            }
                            bit bandwidth {
                                position 1;
                            }
                            bit power {
                                position 2;
                            }
                            bit modulation {
                                position 3;
                            }
                            bit interleaver {
                                position 4;
                            }
                            bit j83annex {
                                position 5;
                            }
                            bit symbol-rate {
                                position 6;
                            }
                            bit mute {
                                position 7;
                            }
                        }
                        default " ";
                        description "This attribute represents the PHY parameters Lock state of the
QAM channels for DEPI-initiated PHY parameters updates.";
                    }
                    leaf allocation-type {
                        type downstream-allocation-type;
                        default any;
                        description "This attribute defines the services this specific DownChannel
instance can be allocated to.";
                    }
                    list encryption-capability {
                        key encryption-capability-index;
                        min-elements 0;
                        max-elements 3;
                        description "The EncryptionCapability object defines one encryption option
of the Encryptor that needs to be reported to the ERM. There can be up to three EncryptionCapability
```

```
objects per QAM. In return, the ERM is expected to create dynamic sessions using one of the reported
encryption options.";
                        leaf encryption-capability-index {
                            type uint32;
                            mandatory true;
                            description "This attributes assigns a unique identifier to this
instance of the EncryptionCapability object.";
                        }
                        leaf ca-encryptor {
                            type ca-encryptor-type-type;
                            mandatory true;
                            description "This enumeration defines the type of CA encryption the
Encryptor uses. The value of other(1) is used when a vendor-extension has been implemented for this
attribute.";
                        }
                        leaf encryption-scheme {
                            type encryption-scheme-type;
                            mandatory true;
                            description "This attribute defines the encryption algorithms applicable
to the CA encryption defined by the CaEncryptor attribute. The value of other(1) is used when a
vendor-extension has been implemented for this attribute.";
                        }
                        leaf key-length {
                            type uint32;
                            mandatory true;
                            description "This attribute defines the key length applicable to the
algorithm defined by the EncryptionScheme attribute.";
                        }
                        container yang-ext {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                    leaf erm-name {
                        type leafref {
                            path "/ccap/video/erm-registration/erm-name";
                        }
                        mandatory true;
                        description "The ERM that manages this down channel.";
                    }
                    container yang-ext5 {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                case yang-choice-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
        }
        grouping video-down-channel-group {
            leaf video-output-tsid {
                type uint16;
                mandatory true;
                description "This attribute specifies the output TSID of the channel. This attribute is
required for a VideoDownChannel. The TSID is globally unique per CCAP. Replicated output streams
share the same OutputTsid.";
            }
        }
        grouping docsis-down-channel-group {
            leaf id {
                type uint32;
                description "Unique identifier per mac-domain for the docsis-down-channel. Used in
DOCSIS messages. A value of 0 (zero) means that the CCAP will automatically assign the Id.";
            }
            leaf provisioned-attribute-mask {
                type attribute-mask-type;
                description "This attribute contains Provisioned Attribute Mask of non-bonded service
flow assignment to this channel.";
            }
            leaf docsis-output-tsid {
                type uint16;
                default 0;
                description "This optional attribute specifies the output TSID of the channel. This
attribute is optional for DocsisDownChannel. The TSID is globally unique per CCAP.  Replicated
output streams share the same OutputTsid.
```

```
";
        }
    }
    grouping ds-rf-port-group {
        leaf rf-mute {
            type boolean;
            default false;
            description "The attribute rf-mute refers to a diagnostic state defined in the DOCSIS RF
Interface (DRFI) Specification. Muting an RF port affects only the output power and does not impact
the operational status of any channel on the port.  ";


            reference "DOCSIS Downstream RF Interface Specification CM-SP-DRFI-I11-110210.";
        }
        leaf base-channel-power {
            type int32;
            units TenthdBmV;
            mandatory true;
            description "This attribute configures the base output power  for each single
DownChannel on the DsRfPort.  The value is expressed in dBmV in units of Tenthdb. The default value
is vendor specific. Acceptable power ranges for this attribute are defined in the Power per Channel
CMTS or EQAM section of [DRFI].
";
            reference "DOCSIS Downstream RF Interface Specification CM-SP-DRFI-I11-110210 Power per
Channel CMTS or EQAM section";
        }
        uses port-group;
        list down-channel {
            key channel-index;
            unique frequency;
            max-elements 158;
            description "A list of down channels.";
            uses down-channel-group;
            choice channel-type {
                mandatory true;
                case docsis {
                        description
                            "The docsis-down-channel object is a down-channel used exclusively for
DOCSIS. The down-channel is its generalization.
                             The docsis-down-channel object is a specialization of down-channel.";
                        uses docsis-down-channel-group;
                        leaf docsis-phy-profile-index {
                            type leafref {
                                path "/ccap/chassis/docsis-phy-profile/phy-index";
                            }
                            description "A reference to a Docsis physical parameters profile. The
default for this index is vendor-dependent; it shall default to an index that is always present in
the phy-profile list.";
                       }
                       container yang-ext1 {
                           ccap:extensionPoint; //different pyang flags impact use of this hint
                           description "node for vendor YANG extensions";
                       }
                }
                case video {
                         description "The video-down-channel object is a down-channel used
exclusively for video channel configuration";
                         uses video-down-channel-group;
                         leaf video-phy-profile-index {
                             type leafref {
                                 path "/ccap/chassis/video-phy-profile/phy-index";
                             }
                             description "A reference to a video physical parameters profile. The
default for this index is vendor-dependent; it shall default to an index that is always present in
the phy-profile list.";
                         }
                         container yang-ext2 {
                             ccap:extensionPoint; //different pyang flags impact use of this hint
                             description "node for vendor YANG extensions";
                         }
                }
                case yang-choice-ext2 {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            container yang-ext {
                ccap:extensionPoint; //different pyang flags impact use of this hint
```

```
                description "node for vendor YANG extensions";
            }
        }
    }
    grouping us-rf-port-group {
        uses port-group;
        list upstream-physical-channel {
            key channel-index;
            unique frequency;
            description
                "The upstream-physical-channel object represents DOCSIS operation on a single
upstream center frequency at a particular channel width.
                 Since CCAP is expected to operate with only DOCSIS 2.0 or later upstream channels,
at least one UpstreamLogicalChannel object (ifType 205) is needed to be instantiated to operate
within an upstream-physical-channel.
                 This object differs from the same object in DOCSIS in that the desired input power
is now set at the upstream-physical-channel and not on a per-LogicalUpstreamChannel instance. If the
target receive power level for an individual logical channel under a physical channel is desired to
be different than the target power level for the physical channel, this can be configured using the
PowerLevelAdjust attribute of the UpstreamLogicalChannel object.";
            leaf channel-index {
                type uint32;
                description "This key identifies a channel on a specific RF Port.";
            }
            leaf admin-state {
                type admin-state-type;
                default down;
                description "This attribute represents the administrative status of the channel. For
downstream channels, Setting the value to down(3) results in the channel being muted, and the value
of testing(4) is used to generate a continuous test wave on this QAM channel.";
            }
            leaf up-down-trap-enabled {
                type boolean;
                default true;
                description "Indicates whether linkUp/linkDown SNMP traps should be generated for
this interface.";
            }
            leaf frequency {
                type uint32 {
                    range "5000000..85000000";
                }
                units Hertz;
                mandatory true;
                description "This attribute configures the center frequency of the upstream-
physical-channel, in Hertz. As of DOCSIS 3.0, the minimum permitted value is the center frequency
such that the lower channel edge is 5000000 Hz and the maximum permitted value is the center
frequency at which the upper channel edge is 85000000 Hz.  This attribute corresponds to the
docsIfUpChannelFrequency object of DOCS-IF-MIB [RFC 4546]. ";
                reference "RFC 4546 DOCS-IF-MIB";
            }
            leaf width {
            type us-channel-width;
                mandatory true;
                description "This attribute configures the width of the upstream-physical-channel,
in Hertz.  The only permitted values as of DOCSIS 3.0 are 1600000, 3200000, and 6400000. The CCAP
may support the narrower channel widths 200000, 400000 and 800000KHz.  This attribute corresponds to
the docsIfUpChannelFrequency object of DOCS-IF-MIB [RFC 4546].";
                reference "RFC 4546 DOCS-IF-MIB";
            }
            leaf power-level {
                type uint32;
                units TenthdBmV;
                mandatory true;
                description "This attribute configures the desired input power level, in TenthdBmV,
common for all upstream logical channels associated with this physical channel instance. The power
level for individual logical channel can be deviated from the common power level through
configuration of PowerLevelAdjust attribute.";
            }
            list upstream-logical-channel {
                key upstream-logical-channel-index;
                description "A list of upstream logical channels.";
                min-elements 1;
                uses upstream-logical-channel-group;
                choice logical-channel-type {
                    mandatory true;
                    case scdma-logical-channel {
                            description
```

```
                                "This configuration object is constructed from the SCDMA fields of
the docsIfUpstreamChannelTable defined in [RFC 4546] and the DOCS-IFEXT2-MIB defined in Annex H of
[OSSIv3.0].
                                 The scdma object is an optional grouping of additional parameters
to an UpstreamLogicalChannel that is defined only for UpstreamLogicalChannel objects that reference
an SCDMA modulation profile.";
                            reference
                                "DOCSIS 3.0 Operations Support System Interface Specification
                                 CM-SP-OSSIv3.0-I15-110623 Requirements for DOCS-IFEXT2-MIB Annex;
RFC 4546 docsIfUpstreamChannelTable.";
                            leaf active-codes {
                                type uint32;
                                description "This attribute configures the number of active codes
used in this channel.  Note that legal values from 64..128 are required to be non-prime. Valid
values are 0|64..66| 68..70| 72| 74..78| 80..82| 84..88| 90..96| 98..100|102| 104..106|108
|110..112|114..126|128.";
                            }
                            leaf codes-per-slot {
                                type uint32;
                                description "Applicable for SCDMA channel types only. The configured
number of SCDMA codes per mini-slot.";
                            }
                            leaf frame-size {
                                type uint32;
                                description "Applicable for SCDMA channel types only. The configured
SCDMA frame size in units of spreading intervals.";
                            }
                            leaf hopping-seed {
                                type uint32;
                                description "Applicable for SCDMA channel types only. 15-bit seed
used for code hopping sequence initialization. Returns zero for non-SCDMA channel types.";
                            }
                            leaf msc-state {
                                type msc-state-type;
                                default channel-disabled;
                                description "Indicates the state of the Maximum Scheduled Codes
feature for an individual logical channel on the CCAP.";
                                reference
                                    "DOCSIS 3.0 Operations Support System Interface Specification
                                     CM-SP-OSSIv3.0-I15-110623 Requirements for DOCS-IFEXT2-MIB
Annex.";
                            }
                            leaf msc-minimum-value {
                                type uint8 {
                                    range "4..128";
                                }
                                units codes;
                                default 4;
                                description "When Maximum Scheduled Codes is enabled, instructs the
CCAP to assign cable modems MSC values no less than this value.";
                            }
                            container yang-ext1 {
                                ccap:extensionPoint; //different pyang flags impact use of this hint
                                description "node for vendor YANG extensions";
                            }
                        }
                        case tdma-logical-channel {
                            description "This configuration object is a specialization of the
docsIfUpstreamChannelTable defined in [RFC 4546] for TDMA logical channels.
";
                            reference
                                "DOCSIS 3.0 Operations Support System Interface Specification
                                 CM-SP-OSSIv3.0-I15-110623 Requirements for DOCS-IFEXT2-MIB Annex;
RFC 4546 docsIfUpstreamChannelTable.";
                            container yang-ext2 {
                                ccap:extensionPoint; //different pyang flags impact use of this hint
                                description "node for vendor YANG extensions";
                            }
                        }
                        case atdma-logical-channel {
                            description "This configuration object is a specialization of the
docsIfUpstreamChannelTable defined in [RFC 4546] for ATDMA logical channels.

";
                            reference
                                "DOCSIS 3.0 Operations Support System Interface Specification
                                 CM-SP-OSSIv3.0-I15-110623 Requirements for DOCS-IFEXT2-MIB Annex;
RFC 4546 docsIfUpstreamChannelTable.";
```

```
                            container yang-ext3 {
                                ccap:extensionPoint; //different pyang flags impact use of this hint
                                description "node for vendor YANG extensions";
                            }
                        }
                        case tdma-and-atdma-logical-channel {
                            description "This configuration object is a specialization of the
docsIfUpstreamChannelTable defined in [RFC 4546] for mixed TDMA/ATDMA logical channels.

";
                            reference
                                "DOCSIS 3.0 Operations Support System Interface Specification
                                 CM-SP-OSSIv3.0-I15-110623 Requirements for DOCS-IFEXT2-MIB Annex;
RFC 4546 docsIfUpstreamChannelTable.";
                            container yang-ext4 {
                                ccap:extensionPoint; //different pyang flags impact use of this hint
                                description "node for vendor YANG extensions";
                            }
                        }
                        case yang-choice-ext {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
        }
        grouping down-channel-phy-params-group {
            leaf phy-index {
                type uint32;
                description "The index of the profile.";
            }
            leaf modulation {
                type qam-modulation-type;
                default qam256;
                description "Defines the modulation type used. The value of other(1) is used when a
vendor-extension has been implemented for this attribute.";
            }
            leaf interleaver-depth {
                type interleaver-depth-type;
                default fecI128J1;
                description
                    "This attribute represents the interleaving depth or operation mode of the
interleaver. The value of other(1) is used when a vendor-extension has been implemented for this
attribute.
                     This attribute is ignored when downstream-phy-standard has a value other than
j83annexB.";
            }
            leaf downstream-phy-standard {
                type downstream-phy-type;
                default j83annexB;
                description "This attribute specifies the standard supported by the QAM channel. A value
of 'dvbc' corresponds to J.83 Annex A. The value of other(1) is used when a vendor-extension has
been implemented for this attribute";
            }
        }
        grouping upstream-logical-channel-group {
            leaf upstream-logical-channel-index {
                type uint32;
                description "The channel index for an upstream logical channel.";
            }
            leaf admin-state {
                type admin-state-type;
                default down;
                description "This attribute configures the administrative state of the upstream logical
channel.";
            }
            leaf up-down-trap-enabled {
                type boolean;
                default false;
```

```
            description "Indicates whether linkUp/linkDown SNMP traps should be generated for this
interface.";
        }
        leaf channel-id {
            type uint32;
            default 0;
            description "This attribute permits an operator to optionally configure the upstream
channel ID signaled in the DOCSIS protocol for the UpstreamLogicalChannel. If zero, CMTSs
automatically assign the DocsisUpChannelId, and this attribute is only read. An operator can create
or update this attribute with a non-zero value to force the CCAP to use the configured DOCSIS
channel ID. A unique configured value exists within the MacDomain to which the upstream-physical-
channel containing this UpstreamLogicalChannel is associated.";
        }
        leaf slot-size {
            type uint32;
            mandatory true;
            description "This attribute configures the number of 6.25 microsecond ticks in each
upstream mini-slot for the UpstreamLogicalChannel. This attribute may have different values for the
different UpstreamLogicalChannel objects on the same upstream-physical-channel. This attribute is
applicable to TDMA and ATDMA channel types only; its value is read and written as zero for SCDMA
type channels.";
        }
        leaf ranging-backoff-start {
            type uint8 {
                range "0..16";
            }
            mandatory true;
            description "This attribute is the initial random back-off window to use when retrying
Ranging Requests. It is expressed as a power of 2. A value of 16 at the CCAP indicates that a
proprietary adaptive retry mechanism is to be used.";
        }
        leaf ranging-backoff-end {
            type uint8 {
                range "0..16";
            }
            mandatory true;
            description "This attribute is the final random back-off window to use when retrying
Ranging Requests. It is expressed as a power of 2. A value of 16 at the CCAP indicates that a
proprietary adaptive retry mechanism is to be used.";
        }
        leaf transmit-backoff-start {
            type uint8 {
                range "0..16";
            }
            mandatory true;
            description "The initial random back-off window to use when retrying transmissions.
Expressed as a power of 2. A value of 16 at the CCAP indicates that a proprietary adaptive retry
mechanism is to be used. See the associated conformance object for write conditions and
limitations.";
        }
        leaf transmit-backoff-end {
            type uint8 {
                range "0..16";
            }
            mandatory true;
            description "The final random back-off window to use when retrying transmissions.
Expressed as a power of 2. A value of 16 at the CCAP indicates that a proprietary adaptive retry
mechanism is to be used. See the associated conformance object for write conditions and
limitations.";
        }
        leaf pre-equalization-enable {
            type boolean;
            mandatory true;
            description "This attribute enables pre-equalization on the UpstreamLogicalChannel when
its value is true, or disables pre-equalization when its value is false.";
        }
        leaf provisioned-attribute-mask {
            type attribute-mask-type;
            mandatory true;
            description "This attribute configures the 32-bit Provisioned Attribute Mask for the
UpstreamLogicalChannel. This is used by a CCAP to control how upstream service flows are assigned to
the UpstreamLogicalChannel.";
        }
        leaf power-level-adjust {
            type uint32;
            units tenthdbm;
            default 0;
```

```
            description "This attribute configures the adjustment from the common power level
configured for the physical US channel in TenthdB. The sum of the PowerLevel and PowerLevelAdjust
determines the expected input power level for the logical channel. If the CCAP does not support the
ability to set the PowerLevelAdjust attribute to a non-zero value, the CCAP may log an error upon
execution of an XML configuration file that contains this attribute value.";
        }
        leaf modulation {
            type leafref {
                path "/ccap/docsis/modulation-profile/modulation-index";
            }
            mandatory true;
            description "A reference to a modulation index.";
        }
    }
    grouping ip-interface-group {
        leaf ip-interface-name {
            type string;
            description "Chassis unique name of the IP interface";
        }
        list primary-ipv4 {
            key ip-address;
            max-elements 1;
            ccap:inlineType;
            description "The primary-ipv4 list allow zero or one primary IPv4 address and prefix for
this IP interface.";
            leaf ip-address {
                type ipv4-host-prefix;
                description "This attribute configures the primary IPv4 address and prefix for this
instance.";
            }
            container yang-ext {
                ccap:extensionPoint; //different pyang flags impact use of this hint
                description "node for vendor YANG extensions";
            }
        }
        list ipv6 {
            key ipv6-address;
            description "The ipv6 list allows zero or more IPv6 interface addresses and prefixes to
be configured for this IP interface";
            ccap:inlineType;
            leaf ipv6-address {
                type ipv6-host-prefix;
                description "This attribute configures the IPv6 address and prefix for this
instance.";
            }
            container yang-ext {
                ccap:extensionPoint; //different pyang flags impact use of this hint
                description "node for vendor YANG extensions";
            }
        }
        list secondary-ipv4 {
            key ip-address;
            description "The secondary-ipv4 list allows zero or more secondary addresses and
prefixes to be configured for this IP interface";
            ccap:inlineType;
            leaf ip-address {
                type ipv4-host-prefix;
                description "This attribute configures the secondary IPv4 address and prefix for
this instance.";
            }
            container yang-ext {
                ccap:extensionPoint; //different pyang flags impact use of this hint
                description "node for vendor YANG extensions";
            }
        }
    }
    grouping enet-port-group
    {
        uses port-group;
        list ip-interface {
            key ip-interface-name;
            max-elements 1;
            ccap:inlineType;
            description "An ip-interface object.";
            uses ip-interface-group;
            leaf ingress-acl {
                type leafref {
                    path "/ccap/network/acl/acl-name";
```

```
                }
            }
            leaf egress-acl {
                type leafref {
                    path "/ccap/network/acl/acl-name";
                }
            }
            container yang-ext {
                ccap:extensionPoint; //different pyang flags impact use of this hint
                description "node for vendor YANG extensions";
            }
        }
    }
    grouping one-gb-ethernet-port-group
    {
        uses enet-port-group;
        leaf duplex {
            type duplex-type;
            default full-duplex;
            description "This attribute configures the Ethernet DuplexState of the interface.";
        }
        leaf speed {
        type ethernet-speed-type;
            mandatory true;
            description "This attribute configures the speed of the interface for interfaces that
can support multiple speeds.";
        }
    }
    grouping epon-one-gb-port-reference {
        description "Reference to a onegigabit epon port.";
        leaf epon-slot {
            type leafref {
                path "/ccap/chassis/slot/slot-number";
            }
            description "Reference to the slot in which the onegigabit epon port resides.";
        }
        leaf epon-port-number {
            type leafref {
                path "/ccap/chassis/slot[slot-number=current()/../epon-slot]/one-gb-epon-port/port-
number";
            }
            description "Reference to a onegigabit epon port number.";
        }
    }
    grouping epon-ten-gb-port-reference {
        description "Reference to a tengigabit epon port.";
        leaf epon-slot {
            type leafref {
                path "/ccap/chassis/slot/slot-number";
            }
            description "Reference to the slot in which the tengigabit epon port resides.";
        }
        leaf epon-port-number {
            type leafref {
                path "/ccap/chassis/slot[slot-number=current()/../epon-slot]/ten-gb-epon-port/port-
number";
            }
            description "Reference to a tengigabit epon port number.";
        }
    }
    grouping udp-map {
        leaf udp-map-index {
            type uint32;
            description "This key represents a globally unique identifier of the UdpMap instance.";
        }
        leaf starting-udp-port {
            type inet:port-number;
            default 0;
            description "This attribute represents the UDP port range start value.";
        }
        leaf port-count {
            type uint32;
            default 0;
            description "This attribute represents the number of  UDP ports starting from
'StartingUdpPort' attribute value.";
        }
    }
    grouping authorizer-group {
```

```
        leaf auth-server-index {
            type uint32;
            description "The index is an unsigned, 32-bit identifier used as a key for this object
instance.";
        }
        container auth-server {
            uses host;
            description "This attribute is the IP address/hostname of the RADIUS server referred to
in this table entry.";
        }
        leaf auth-key {
            type string;
            mandatory true;
            description "This attribute corresponds to the shared secret that is used to encrypt the
communication. Upon export, the CCAP shall export the Key attribute of the object encrypted with a
vendor-specific algorithm.";
        }
        leaf auth-clear-key {
            type boolean;
            mandatory true;
            description "This attribute indicates whether the Key attribute is included in the XML
configuration file in the clear (true) or encrypted (false). This attribute defines the status of
the key (encrypted or decrypted), not whether the device should export the key in the clear or
encrypted. Regardless of the value of this setting, the key will always be exported as encrypted.";
        }
        leaf auth-timeout {
            type uint32 {
                range "1..4294967295";
            }
            units seconds;
            default 3;
            description "This attribute defines the number of seconds before a connection is
declared inactive.";
        }
        leaf auth-retransmit-attempts {
            type uint32;
            units "Number of retransmissions";
            default 1;
            description "This attribute defines the number of retransmissions before giving up the
connection.";
        }
        leaf primary-auth-server {
            type boolean;
            default false;
            description "This attribute designates whether this auth instance is the primary or
backup server. If multiple instances are set to false, when the primary connection fails, the
selection of which backup server to use is vendor specific.";
        }
        leaf source-ip-interface-name {
            type string;
            description "Local IP interface; needs to be a configured ip-interface name. If not
specified, then the vendor picks the source IP address";
        }
    }
    grouping terminal-service-group {
        leaf screen-length {
            type uint16;
            default 24;
        }
        leaf screen-width {
            type uint16;
            default 80;
        }
        container input-transport-controls {
            leaf telnet-enabled {
                type boolean;
                default false;
            }
            leaf ssh-enabled {
                type boolean;
                default false;
            }
            container yang-ext {
                ccap:extensionPoint; //different pyang flags impact use of this hint
                description "node for vendor YANG extensions";
            }
        }
    }
```

```
    grouping ssm-group {
        leaf source-prefix-address {
            type inet:ip-prefix;
            mandatory true;
            description "This attribute identifies a specific Multicast Source prefix defined for
this rule.
                         A source prefix of length 32 is a host source address.
                         A Source prefix of length zero is defined as 'all source addresses' (*, G).
                         Source prefix addresses are unicast addresses.";
            reference "RFC 3569 section 6; RFC 3306 sections 5 and 6.";
        }
        leaf group-prefix-address {
            type inet:ip-prefix;
            mandatory true;
            description "This attribute is the IP prefix corresponding to an IP multicast group.";
        }
    }
    grouping ecm-group {
        leaf ecm-index {
            type uint32;
            description "The Index is an unsigned, 32-bit identifier used as a key for this
object.";
        }
        container ecm-server {
            uses host;
            description "This is the IP address/hostname of the ECM server.";
        }
        leaf ecm-server-port {
            type inet:port-number;
            mandatory true;
            description "This is the far-end TCP port for communicating with the ECM server.";
        }
        leaf ecm-cas-id {
            type ccap-octet-data-type {
               length "8";
            }
            mandatory true;
            description "This attribute defines the Ca System Id of the ECM server.";
        }
    }
    grouping video-session-group {
        leaf session-index {
            type uint32;
        }
        leaf session-name {
            type string;
            default " ";
            description "This attribute configures a name for the PID session. ";
        }
        leaf session-input-ts {
            mandatory true;
            type leafref {
                path "/ccap/video/video-input-ts/input-ts-index";
            }
            description "A reference to an input-ts-index.";
        }
        list session-output-ts {
          key session-output-ts-index;
          min-elements 1;
          description "A reference to an output-ts object.";
          leaf session-output-ts-index {
            type leafref {
              path "/ccap/video/video-output-ts/output-ts-index";
            }
            description "A reference to an output-ts-index.";
          }
          container yang-ext {
              ccap:extensionPoint; //different pyang flags impact use of this hint
              description "node for vendor YANG extensions";
          }
       }
    }
    grouping virtual-interface-group {
        leaf interface-index {
            type uint8;
            mandatory true;
            description "The index for this virtual ip-interface";
           }
```

```
        leaf admin-state {
            type admin-state-type;
            default down;
            description "This attribute configures the administrative state of the virtual
interface.";
        }
        list ip-interface {
            key ip-interface-name;
            max-elements 1;
            ccap:inlineType;
            description "An ip-interface object.";
            uses ip-interface-group;
            container yang-ext {
                ccap:extensionPoint; //different pyang flags impact use of this hint
                description "node for vendor YANG extensions";
            }
        }
    }
    grouping view-config-ref {
        list view-config-ref {
            key view-name;
            leaf view-name {
                type leafref {
                    path "/ccap/management/snmp/view-config/view-name";
                }
            }
            min-elements 0;
            description "A reference to a view-configuration name.";
            container yang-ext {
                ccap:extensionPoint; //different pyang flags impact use of this hint
                description "node for vendor YANG extensions";
            }
        }
    }
    grouping acl-filter-group {
        choice protocol-type {
            case protocol-value {
                    ccap:inlineType;
                    leaf protocol-id {
                        type uint8;
                        mandatory true;
                    }
                    container yang-ext1 {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
            }
            case icmp {
                    ccap:inlineType;
                    leaf icmp-message-type {
                        type uint16;
                        mandatory true;
                    }
                    leaf icmp-message-code {
                        type uint16;
                    }
                    container yang-ext2 {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
            }
            case igmp {
                container igmp {
                    ccap:inlineType;
                    leaf igmp-message-type {
                        type uint16;
                        mandatory true;
                    }
                    container yang-ext3 {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
            }
            case yang-protocol-choice-ext {
                ccap:extensionPoint; //different pyang flags impact use of this hint
                description "node for vendor YANG extensions";
            }
```

```
    }
    leaf fragments {
        type boolean;
        default false;
    }
    choice source-port {
        case source-port-range {
                ccap:inlineType;
                leaf start-sport {
                    type inet:port-number;
                    mandatory true;
                }
                leaf end-sport {
                    type inet:port-number;
                    mandatory true;
                }
                container yang-ext7 {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
        }
        case single-source-port {
                ccap:inlineType;
                leaf sport {
                    type inet:port-number;
                    mandatory true;
                }
                leaf sport-comparator {
                    type acl-comparator-type;
                    default eq;
                }
                container yang-ext8 {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
        }
        case yang-sport-choice-ext {
            ccap:extensionPoint; //different pyang flags impact use of this hint
            description "node for vendor YANG extensions";
        }
    }
    choice dest-port {
        case dest-portrange {
                ccap:inlineType;
                leaf start-dport {
                    type inet:port-number;
                    mandatory true;
                }
                leaf end-dport {
                    type inet:port-number;
                    mandatory true;
                }
                container yang-ext9 {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
        }
        case single-dest-port {
                ccap:inlineType;
                leaf dport {
                    type inet:port-number;
                    mandatory true;
                }
                leaf dport-comparator {
                    type acl-comparator-type;
                    default eq;
                }
                container yang-ext10 {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
        }
        case yang-dport-choice-ext {
            ccap:extensionPoint; //different pyang flags impact use of this hint
            description "node for vendor YANG extensions";
        }
    }
}
```

```
    grouping ccap-group {
        reference
            "CCAP Operations Support System Interface Specification
             CM-SP-CCAP-OSSI-I01-110930 Ccap Object section.";
        leaf name {
            type string {
                length "1..32";
            }
            description "This attribute defines the name of the CCAP platform being configured.";
        }
        leaf description {
            type string;
            description "This attribute contains the description of the CCAP platform.";
        }
        leaf location {
            type string {
                length "1..128";
            }
            description "This attribute contains any location information for the CCAP.";
        }
        uses vendor-extension-version;
        container chassis {
            description "The Chassis container allows the user to configure the CCAP hardware
elements.";
            list decryptor {
                key decryptor-index;
                description "The Decryptor object provides for the configuration of a Decryptor
module or modules in the CCAP that are used to decrypt encrypted content delivered to the CCAP.  ";
                leaf decryptor-index {
                    type uint32;
                    description "The Index is an unsigned, 32-bit identifier used as a key for this
object.";
                }
                leaf cw-timeout {
                    type uint32;
                    units seconds;
                    default 10;
                    description "This attribute configures the length of time in seconds that the
decrypted should wait for an ECMD before switching to a redundant unit. ";
                }
                list ecmd-usage {
                    key ecmd-usage-index;
                    min-elements 1;
                    description "The ecmd-usage object provides for the configuration of multiple
decryption sessions. It is an intermediate object that provides linkages between Decryptor objects
and the ECMD(s) associated with those encrypted streams.";


                    leaf ecmd-usage-index {
                        type uint32;
                        description "This is an index for an instance of this object. The ecmd-usage
object is a pointer to an ECMD that can be used for any program session that requires decryption as
long as the CAS identifier of the input program matches.";
                    }
                    leaf priority {
                        type uint32;
                        mandatory true;
                        description "This is the configured selection priority for any program
session that requires decryption when multiple ECMDs with the same CAS identifier are active. The
ECMD with the lowest number should be selected first.";
                    }
                    leaf ecmd-ref {
                        type leafref {
                            path "/ccap/video/ecmd/ecm-index";
                        }
                        description "A reference to an instance of an ecmd object, referenced by its
ecm-index.";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
            }
            list fiber-node-config {
                key fiber-node-config-index;
                description
```

```
                    "Fiber-node-config defines the cable hybrid fiber/coax system (HFC) plant Fiber
Nodes reached by RF ports on a CCAP.
                     This object supports the creation and deletion of multiple instances.";
                reference
                    "DOCSIS 3.0 Operations Support System Interface Specification
                     CM-SP-OSSIv3.0-I15-110623 FiberNodeCfg Object section.";
                leaf fiber-node-config-index {
                    type uint32;
                    description "The index of the fiber node being configured.";
                }
                leaf fiber-node-name {
                    type string {
                        length "1..64";
                    }
                    mandatory true;
                    description "This key represents a human-readable name for a fiber node. ";
                    reference
                        "DOCSIS 3.0 MAC and Upper Layer Protocols Interface Specification
                         CM-SP-MULPIv3.0-I16-110623 RF Topology Configuration section.";
                }
                leaf description {
                    type string;
                    default " ";
                    description "This attribute represents a human-readable description of the
node.";
                }
                list ds-rf-port-ref {
                    key "slot ds-rf-port";
                    description "This object associates a downstream RF port with this fiber node
configuration instance.";
                    uses ds-rf-port-reference;
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list us-rf-port-ref {
                    key "slot us-rf-port";
                    description "This object associates an upstream RF port with this fiber node
configuration instance.";
                    uses us-rf-port-reference;
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list slot {
                key slot-number;
                description "This object configures a slot within the CCAP chassis. Line cards
reside in slots.";
                reference
                    "CCAP Operations Support System Interface Specification
                     CM-SP-CCAP-OSSI-I01-110930 Slot section.";
                leaf slot-number {
                    type uint8;
                    description "This attribute configures the slot number for which a LineCard
object will be configured. The Number attribute is a zero- or one-based index that sequentially
numbers the physical slots in the chassis. For example, the Slot numbers start at zero and increase
to n-1, where n is the number of slots the chassis supports.";
                }
                choice line-card-type {
                    mandatory true;
                    case rf-line-card {
                            description "This is an RF line card that can have downstream RF ports,
upstream RF ports, or both.";
                            reference
                                "CCAP Operations Support System Interface Specification
                                 CM-SP-CCAP-OSSI-I01-110930 UsDsRfLineCard section.";
                            container rf-card {
                                uses line-card-group;
                            }
                            list encryptor {
                                key encryptor-index;
```

```
                                description
                                    "This object allows for the configuration of an Encryptor. Each
Encryptor
                                     object is part of a DLC. Each is associated with at
                                     least one active and zero or more backup ECMGs. For
                                     Simulcrypt, the Encryptor would be associated with
                                     multiple active ECMGs, each for a different
                                     CAS. Each is also associated with one or more video
                                     sessions that is being encrypted on this DLC.";
                                uses encryptor-group;
                                container yang-ext {
                                    ccap:extensionPoint; //different pyang flags impact use of this
hint
                                    description "node for vendor YANG extensions";
                                }
                            }
                            leaf enable-udp-map-encryption {
                                type leafref {
                                    path "/ccap/video/static-udp-map-encryption/udp-map-encryption-
index";
                                }
                                description "Reference to an encryptor for all statically udp port
mapped sessions on this card.
                                             If not present, no statically udp port mapped video
session is encrypted locally.";
                            }
                            list us-rf-port {
                                key port-number;
                                description
                                    "A us-rf-port object represents a physical upstream RF connector
on a CCAP
                                     line card.  It is derived from the Port abstract
                                     class, and so inherits all attributes of that
                                     class, including its associations.  A UsRfPort may
                                     be contained by either a UsRfLineCard or a
                                     UsDsRfLineCard. It contains one or more
                                     upstream-physical-channels. This object has no
                                     attributes other than what has been inherited from
                                     the abstract class Port, but does have several
                                     associations.";
                                uses us-rf-port-group;
                                container yang-ext {
                                    ccap:extensionPoint; //different pyang flags impact use of this
hint
                                    description "node for vendor YANG extensions";
                                }
                            }
                            list ds-rf-port {
                                key port-number;
                                description
                                    "This object allows for the configuration of a physical
Downstream RF
                                     port on a DLC or UsDsRfLineCard.  The DsRfPort is a
                                     type of the abstract class Port and inherits those
                                     common parameters.  In the CCAP, a single port now
                                     encompasses the entire downstream spectrum instead
                                     of a few carriers as are seen in the current
                                     generation EQAM and CMTS products.";
                                uses ds-rf-port-group;
                                container yang-ext {
                                    ccap:extensionPoint; //different pyang flags impact use of this
hint
                                    description "node for vendor YANG extensions";
                                }
                            }
                            container yang-ext1 {
                                ccap:extensionPoint; //different pyang flags impact use of this hint
                                description "node for vendor YANG extensions";
                            }
                        }
                        case epon-line-card {
                            description "This object configures an EPON line card. An epon-line-card
that only supports PON Ports of type one-gb-epon-port will not have a configured enet-port. An epon-
line-card that supports PON ports of type ten-gb-epon-port may have a configured enet-port.";
                            container epon-card {
                                uses line-card-group;
                            }
                            list one-gb-epon-port {
```

```
                                key port-number;
                                description "This configuration object allows for a one Gigabit EPON
port to be configured on an EPON line card. It is a type of the abstract class PonPort.";
                                uses port-group;
                                leaf upstream-fec-mode {
                                    type upstream-fec-mode-type;
                                    default disabled;
                                    description
                                         "This attribute configures the FEC mode applied to the EPON
upstream. The perOnu option allows the ONU provisioning process to determine whether FEC should be
enabled or disabled. This option is only valid for 1G EPON interfaces.
                                         The default value for the 1G EPON interface would be
'disabled'.
                                         The default value for the 10G EPON interface would be
'enabled'.
                                         The value of other(1) is used when a vendor-extension has
been implemented for this attribute.";
                                }
                                leaf downstream-fec-mode {
                                    type downstream-fec-mode-type;
                                    default disabled;
                                    description
                                        "This attribute configures the FEC mode of the EPON
downstream. The perOnu option allows the ONU provisioning process to determine whether FEC should be
enabled or disabled. This option is only valid for 1G EPON interfaces.
                                         The default value for the 1G EPON interface would be
'disabled'.
                                         The default value for the 10G EPON interface would be
'enabled'.
                                         The value of other(1) is used when a vendor-extension has
been implemented for this attribute.";
                                }
                                container yang-ext {
                                    ccap:extensionPoint; //different pyang flags impact use of this
hint
                                    description "node for vendor YANG extensions";
                                }
                            }
                            list ten-gb-epon-port {
                                key port-number;
                                description "This configuration object allows for a symmetric or
asymmetric ten Gigabit EPON port to be configured on an EPON line card. It is a type of the abstract
class PonPort";
                                leaf pon-type {
                                    type pon-symmetry-type;
                                    mandatory true;
                                    description "This attribute configures the speed of the 10G EPON
interfaces on the line card and allows for asymmetrical upstream and downstream speeds.";
                                }
                                uses port-group;
                                leaf upstream-fec-mode {
                                    type upstream-fec-mode-type;
                                    default enabled;
                                    description
                                         "This attribute configures the FEC mode applied to the EPON
upstream. The perOnu option allows the ONU provisioning process to determine whether FEC should be
enabled or disabled. This option is only valid for 1G EPON interfaces.
                                         The default value for the 1G EPON interface would be
'disabled'.
                                         The default value for the 10G EPON interface would be
'enabled'.
                                         The value of other(1) is used when a vendor-extension has
been implemented for this attribute.";
                                }
                                leaf downstream-fec-mode {
                                    type downstream-fec-mode-type;
                                    default enabled;
                                    description
                                        "This attribute configures the FEC mode of the EPON
downstream. The perOnu option allows the ONU provisioning process to determine whether FEC should be
enabled or disabled. This option is only valid for 1G EPON interfaces.
                                         The default value for the 1G EPON interface would be
'disabled'.
                                         The default value for the 10G EPON interface would be
'enabled'.
                                         The value of other(1) is used when a vendor-extension has
been implemented for this attribute.";
                                }
```

```
                                container yang-ext {
                                    ccap:extensionPoint; //different pyang flags impact use of this
hint
                                    description "node for vendor YANG extensions";
                                }
                            }
                            list one-gb-ethernet-epon-port {
                                key port-number;
                                description "This object configures a one gigabit interface for an
Ethernet port. The speed and duplex settings for this type of port can be configured via this
object.";
                                uses one-gb-ethernet-port-group;
                                container yang-ext {
                                    ccap:extensionPoint; //different pyang flags impact use of this
hint
                                    description "node for vendor YANG extensions";
                                }
                            }
                            list ten-gb-ethernet-epon-port {
                                key port-number;
                                description "This object configures a ten gigabit interface for an
Ethernet port.";
                                uses enet-port-group;
                                container yang-ext {
                                    ccap:extensionPoint; //different pyang flags impact use of this
hint
                                    description "node for vendor YANG extensions";
                                }
                            }
                            list forty-gb-ethernet-epon-port {
                                key port-number;
                                description "This object configures a 40 gigabit interface for an
Ethernet port.";
                                uses enet-port-group;
                                container yang-ext {
                                    ccap:extensionPoint; //different pyang flags impact use of this
hint
                                    description "node for vendor YANG extensions";
                                }
                            }
                            list one-hundred-gb-ethernet-epon-port {
                                key port-number;
                                description "This object configures a 100 gigabit interface for an
Ethernet port.";
                                uses enet-port-group;
                                container yang-ext {
                                    ccap:extensionPoint; //different pyang flags impact use of this
hint
                                    description "node for vendor YANG extensions";
                                }
                            }
                            container yang-ext2 {
                                ccap:extensionPoint; //different pyang flags impact use of this hint
                                description "node for vendor YANG extensions";
                            }
                    }
                    case sre-line-card {
                            description "The sre-line-card is the name given to the line card in the
integrated CCAP chassis that contains all the NSI and Management functions for the CCAP. This line
card is associated with EnetPort. This object inherits a number of attributes from the LineCard
abstract object.";
                            container sre-card {
                                uses line-card-group;
                            }
                            list one-gb-ethernet-port {
                                key port-number;
                                description "This object configures a one gigabit interface for an
Ethernet port. The speed and duplex settings for this type of port can be configured via this
object.";
                                uses one-gb-ethernet-port-group;
                                container yang-ext {
                                    ccap:extensionPoint; //different pyang flags impact use of this
hint
                                    description "node for vendor YANG extensions";
                                }
                            }
                            list ten-gb-ethernet-port {
                                key port-number;
```

```
                                description "This object configures a ten gigabit interface for an
Ethernet port.";
                                uses enet-port-group;
                                container yang-ext {
                                    ccap:extensionPoint; //different pyang flags impact use of this
hint
                                    description "node for vendor YANG extensions";
                                }
                            }
                            list forty-gb-ethernet-port {
                                key port-number;
                                description "This object configures a 40 gigabit interface for an
Ethernet port.";
                                uses enet-port-group;
                                container yang-ext {
                                    ccap:extensionPoint; //different pyang flags impact use of this
hint
                                    description "node for vendor YANG extensions";
                                }
                            }
                            list one-hundred-gb-ethernet-port {
                                key port-number;
                                description "This object configures a 100 gigabit interface for an
Ethernet port.";
                                uses enet-port-group;
                                container yang-ext {
                                    ccap:extensionPoint; //different pyang flags impact use of this
hint
                                    description "node for vendor YANG extensions";
                                }
                            }
                            container yang-ext3 {
                                ccap:extensionPoint; //different pyang flags impact use of this hint
                                description "node for vendor YANG extensions";
                            }
                        }
                        case yang-choice-ext {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list video-phy-profile {
                    key phy-index;
                    description "The video-phy-profile list is a specialization of the down-channel-phy-
params object. It specifies values for video PHY parameters when those parameters are not set at the
channel level. At least one entry shall always be present and not be deletable. The index of such
entries is vendor specific. While one element needs to be defined for each vendor's list, it is not
required that this element be configured (i.e. it could be defaulted), so min-elements is zero, not
one";
                    reference
                        "CCAP Operations Support System Interface Specification
                         CM-SP-CCAP-OSSI-I01-110930 DrfCfg section.";
                    uses down-channel-phy-params-group;
                    leaf spectrum-inversion {
                        type boolean;
                        default false;
                        description "This attribute specifies global default RF Signal Spectrum
inversion. When set to 'true', it indicates that the QAM channel spectrum is inverted.";
                    }
                    leaf symbol-rate-override {
                        type uint32;
                        description "Sets the symbol rate for the video channel. Video channels can have
more than one symbol rate for a single modulation, unlike DOCSIS channels. If not specified,
channels configured to use this profile operate with the value specified by DOCSIS for the Annex and
modulation.";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list docsis-phy-profile {
                    key phy-index;
```

```
                description "The docsis-phy-profile list is a specialization of the down-channel-
phy-params object. It specifies values for video PHY parameters when those parameters are not set at
the channel level. At least one entry shall always be present and not be deletable. The index of
such entries is vendor specific. While one element needs to be defined for each vendor's list, it is
not required that this element be configured (i.e. it could be defaulted), so min-elements is zero,
not one";
                reference
                    "CCAP Operations Support System Interface Specification
                     CM-SP-CCAP-OSSI-I01-110930 DrfCfg section.";
                uses down-channel-phy-params-group;
            }
            container yang-ext {
                ccap:extensionPoint; //different pyang flags impact use of this hint
                description "node for vendor YANG extensions";
            }
        }
        container docsis {
            description "The docsis object is the primary container of DOCSIS configuration
objects.";
            reference
                "CCAP Operations Support System Interface Specification
                 CM-SP-CCAP-OSSI-I01-110930 DOCSIS Configuration Objects section.";
            choice global-params {
                case docs-global {
                    description "The docs-global object is the container for the DOCSIS global
configuration objects.";
                    leaf maximum-scheduled-codes-enabled {
                        type boolean;
                        mandatory true;
                        description "Indicates the global state of the Maximum Scheduled Code
feature on the CCAP. The value true indicates that this feature can be enabled on individual logical
channels on the CCAP. The value false indicates that the feature is not in operations on the CCAP.
Note that the CCAP Object attribute ScdmaChannelMscState enables or disables Maximum Scheduled Codes
on a per logical channel basis.
    ";
                    }
                    leaf l2-vpn-global-enabled {
                        type boolean;
                        default false;
                        description "This attribute will enable or disable on a global basis the
configuration of L2VPN forwarding for all DOCSIS MAC Domains. The default value is false. This
attribute only enables L2VPN forwarding; configuration of the feature is handled in a vendor-
specific way.";
                    }
                    container yang-ext1 {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
            }
            list cm-vendor-oui {
                key cm-oui;
                min-elements 0;
                description "Cm-vendor-oui allows the operator to create a database of OUIs and
Vendors.";
                leaf cm-oui {
                    type ccap-octet-data-type {
                        length "6";
                    }
                    description "This attribute configures the OUI portion of a given MAC address.";
                }
                leaf cm-vendor-name {
                    type string;
                    mandatory true;
                    description "This attribute configures the company name of the vendor with the
associated OUI.";
                    }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            container docs-security {
                description "The docs-security container is the primary container of DOCSIS security
configuration objects.";
                list sav-config-list {
                    key "sav-config-list-name";
                    min-elements 0;
```

```
                    description
                        "A sav-config-list node consists of read-write objects and is based on the
docsSecSavCfgListEntry defined in [OSSIv3.0] and will be used with no modifications for CCAP. The
RowStatus attribute has been removed.
                         This object supports the creation and deletion of multiple instances. Each
object instance defines one CMTS configured subnet prefix extension for which the CCAP performs
source address verification.
                         Creation of a new instance of this object requires the PrefixAddrType and
PrefixAddr attributes to be set.";
                    reference
                        "DOCSIS 3.0 Operations Support System Interface Specification
                         CM-SP-OSSIv3.0-I15-110623 DOCS-SEC-MIB section.";
                    leaf sav-config-list-name {
                        type string;
                        description "This attribute is the key that identifies the instance of the
SavCmAuth object to which this object extension belongs.";
                    }
                    list sav-rule {
                        key "rule-id";
                    min-elements 1;
                    description "The rules for a particular sav-config-list.";
                    leaf rule-id {
                        type uint32;
                        description "This attribute is the key that identifies a particular subnet
prefix rule of an instance of this object.";
                    }
                    leaf prefix-address {
                        type inet:ip-prefix;
                        mandatory true;
                        description "This attribute corresponds to the IP address and prefix of this
subnet prefix rule.";
                    }
                        container yang-ext {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                container cmts-sav-control {
                    description "Cmts-sav-control container defines attributes for global Source
Address Verification (SAV) configuration.
";
                    reference
                        "DOCSIS 3.0 Operations Support System Interface Specification
                         CM-SP-OSSIv3.0-I15-110623 DOCS-SEC-MIB section.";
                    leaf cm-authentication-enable {
                        type boolean;
                        default true;
                        description
                            "This attribute enables or disables Source Address Verification (SAV)
for CM configured policies in the SavCmAuth object. If this attribute is set to 'false', the CM
configured policies in the SavCmAuth object are ignored.
                             This attribute is only applicable when the SrcAddrVerificationEnabled
attribute of the MdCfg object is 'true'.";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                choice server {
                    case cmts-server-config {
                        description "Cmts-server-config container defines attributes for configuring
TFTP Configuration File Security features.
    ";
                        reference
                            "DOCSIS 3.0 Operations Support System Interface Specification
                             CM-SP-OSSIv3.0-I15-110623 DOCS-SEC-MIB section.";
                        leaf tftp-options {
                            type bits {
                                bit hw-addr {
                                    position 0;
                                }
                                bit net-addr {
```

```
                                position 1;
                            }
                        }
                        mandatory true;
                        description
                            "This attribute instructs the CMTS to insert the source IP address
and/or MAC address of received TFTP packets into the TFTP option fields before forwarding the
packets to the Config File server.
                             This attribute is only applicable when the TftpProxyEnabled
attribute of the MdCfg object is 'true'.";
                    }
                    leaf config-file-learning-enabled {
                        type boolean;
                        default true;
                        description
                            "This attribute enables and disables Configuration File Learning
functionality.
                             If this attribute is set to 'true' the CMTS will respond with
Authentication Failure in the REG-RSP message when there is a mismatch between learned config file
parameters and REG-REQ parameters. If this attribute is set to 'false', the CMTS will not execute
config file learning and mismatch check.
                             This attribute is only applicable when the TftpProxyEnabled
attribute of the MdCfg object is 'true'.";
                    }
                    container yang-ext2 {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
            }
            container cmts-encrypt {
                description "Cmts-encrypt container includes an attribute which defines the
order in which encryption algorithms are to be applied.
";
                reference
                    "DOCSIS 3.0 Operations Support System Interface Specification
                     CM-SP-OSSIv3.0-I15-110623 DOCS-SEC-MIB section.";
                leaf encrypt-alg-priority {
                    type snmp-tag-list-type;
                    default "aes128CbcMode des56CbcMode des40CbcMode";
                    description
                        "This attribute allows for configuration of a prioritized list of
encryption algorithms the CMTS will use when selecting the primary SAID encryption algorithm for a
given CM. The CMTS selects the highest priority encryption algorithm from this list that the CM
supports. By default the following encryption algorithms are listed from highest to lowest priority
(left being the highest): 128 bit AES, 56 bit DES, 40 bit DES.
                         An empty list indicates that the CMTS attempts to use the latest and
most robust encryption algorithm supported by the CM. The CMTS will ignore unknown values or
unsupported algorithms.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            container cmts-certificate {
                description "Cmts-certificate container defines attributes for global
certificate revocation configuration.";
                reference
                    "DOCSIS 3.0 Operations Support System Interface Specification
                     CM-SP-OSSIv3.0-I15-110623 DOCS-SEC-MIB section.";
                leaf cert-revocation-method {
                    type cert-revocation-method-type;
                    default none;
                    description
                        "This attribute identifies which certificate revocation method is to be
used by the CMTS to verify the cable modem certificate validity.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list cmts-cert-revocation-list {
                key url;
                description
```

```
                        "Cmts-cert-revocation-list consists of the read-write objects under the
CmtsCertRevocationList object identifier defined in [OSSIv3.0]. The LastUpdate attribute has been
removed.
                         This object defines a CRL location URL and periodic refresh interval value.
The CRL location URL defines from where the CCAP will retrieve the CRL file. The periodic refresh
interval value indicates how often the CCAP will retrieve the CRL file for updates if the
tbsCertList.nextUpdate attribute in the file is absent.
                         This object is only applicable when the CertRevocationMethod attribute of
the DocsSecCmtsCertificate object is set to 'crl' or 'crlAndOcsp'.";
                    reference
                        "DOCSIS 3.0 Operations Support System Interface Specification
                         CM-SP-OSSIv3.0-I15-110623 DOCS-SEC-MIB section.";
                    leaf url {
                        type string;
                        mandatory true;
                        description "This attribute contains the URL from where the CMTS will
retrieve the CRL file. When this attribute is set to a URL value different from the current value,
it triggers the CMTS to retrieve the CRL file from that URL. If the value of this attribute is a
zero-length string, the CMTS does not attempt to retrieve the CRL.
";
                        reference
                            "DOCSIS 3.0 Security Specification CM-SP-SECv3.0-I13-100611,
                             BPI+ X.509 Certificate Profile and Management section.";
                    }
                    leaf refresh-interval {
                        type uint32 {
                            range "1..524160";
                        }
                        units minutes;
                        default 10080;
                        description "This attribute contains the refresh interval for the CMTS to
retrieve the CRL (referred to in the Url attribute) with the purpose of updating its Certificate
Revocation List. This attribute is meaningful if the tbsCertList.nextUpdate attribute does not exist
in the last retrieved CRL, otherwise the value 0 is returned.
";
                        reference
                            "DOCSIS 3.0 Security Specification CM-SP-SECv3.0-I13-100611,
                             BPI+ X.509 Certificate Profile and Management section.";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list cmts-cm-eae-exclusion {
                    key cmts-cm-eae-exclusion-id;
                    description
                        "This configuration object consists of the read-write objects of the
docsSecCmtsCmEaeExclusion object is defined in [OSSIv3.0] and will be used with no further
modifications for CCAP. The RowStatus attribute has been removed.
                         This object defines a list of CMs or CM groups to exclude from Early
Authentication and Encryption (EAE). This object allows overrides to the value of EAE Control for
individual CMs or group of CMs for purposes such as debugging.


                         The CCAP shall support a minimum of 30 instances of the CmtsCmEaeExclusion
object.
                         This object is only applicable when the EarlyAuthEncryptCtrl attribute of
the MdCfg object is enabled.
                         This object supports the creation and deletion of multiple instances.";
                    reference
                        "DOCSIS 3.0 Operations Support System Interface Specification
                         CM-SP-OSSIv3.0-I15-110623 DOCS-SEC-MIB section.";
                    leaf cmts-cm-eae-exclusion-id {
                        type uint32 {
                            range "1..4294967295";
                        }
                        description "This key uniquely identifies the exclusion MAC address rule.";
                    }
                    leaf mac-address {
                        type yang:mac-address;
                        default 00:00:00:00:00:00;
                        description "This attribute identifies the CM MAC address. A match is made
when a CM MAC address bitwise ANDed with the MacAddrMask attribute equals the value of this
attribute.";
                    }
                    leaf mac-address-mask {
                        type yang:mac-address;
```

```
                        default FF:FF:FF:FF:FF:FF;
                        description "This attribute identifies the CM MAC address mask and is used
with the MacAddr attribute.";
                    }
                    container yang-ext5 {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list cmts-online-cert-status-protocol {
                    key url;
                    max-elements 1;
                    description
                        "This configuration object is defined in [OSSIv3.0] and will be used with no
modifications for CCAP.
                         This object contains an OCSP Responder URL and an attribute to bypass
signature checking of the OCSP response, as detailed in [RFC 2560]. The CCAP will use the URL for
OCSP communications in checking a certificate's revocation status. This object is only applicable
when the CertRevocationMethod attribute of the CmtsCertificate object is set to 'ocsp' or
'crlAndOcsp'.";
                    reference
                        "DOCSIS 3.0 Operations Support System Interface Specification
                         CM-SP-OSSIv3.0-I15-110623 DOCS-SEC-MIB section.";
                    leaf url {
                        type string;
                        mandatory true;
                        description "This attribute contains the URL string to retrieve OCSP
information. If the value of this attribute is a zero-length string, the CMTS does not attempt to
request the status of a CM certificate.
";
                        reference
                            "DOCSIS 3.0 Security Specification CM-SP-SECv3.0-I13-100611,
                             BPI+ X.509 Certificate Profile and Management section; RFC 2560.";
                    }
                    leaf signature-bypass {
                        type boolean;
                        default false;
                        description "This attribute enables or disables signature checking on OCSP
response messages. ";
                        reference
                            "DOCSIS 3.0 Security Specification CM-SP-SECv3.0-I13-100611,
                             BPI+ X.509 Certificate Profile and Management section; RFC 2560.";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                choice sys-bpi {
                    case sys-bpi-config {
                        description "This object provides the configuration of the system-wide
default Baseline Privacy key lifetimes. If not configured, the default values are vendor specific.";
                        reference "RFC 3083";
                        leaf sys-default-authentication-lifetime {
                            type uint32;
                            units seconds;
                            mandatory true;
                            description "The value of this object is the default lifetime, in
seconds, the CCAP assigns to a new authorization key.";
                        }
                        leaf sys-default-tek-lifetime {
                            type uint32;
                            units seconds;
                            mandatory true;
                            description "The value of this object is the default lifetime, in
seconds, the CCAP assigns to a new Traffic Encryption Key (TEK).";
                        }
                        container yang-ext3 {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
```

```
            container docs-subscriber-management {
                description "This group of configuration elements allows for the configuration of
the Subscriber Management rules. They are based on the configuration elements from [OSSIv3.0]";
                container base {
                    description "This object defines the configuration parameters of Subscriber
Management features for the CM in case the CM does not signal any of the parameters during the
registration process.

";
                    reference
                        "DOCSIS 3.0 Operations Support System Interface Specification
                         CM-SP-OSSIv3.0-I15-110623 Base Object section.";
                    leaf cpe-max-ipv4 {
                        type uint16 {
                            range "0..1023";
                        }
                        default 16;
                        description "This attribute represents the maximum number of IPv4 addresses
allowed for the CM's CPE if not signaled in the registration process.";
                    }
                    leaf cpe-max-ipv6 {
                        type uint16 {
                            range "0..1023";
                        }
                        default 16;
                        description "This attribute represents the maximum number of IPv6 Prefixes
and addresses allowed for the CM's CPEs if not signaled in the registration process. All IPv6
prefixes and addresses, including Link-Local and any address with a scope greater than 1 are counted
against the CpeMax Ipv6AddressesDef.";
                    }
                    leaf cpe-active {
                        type boolean;
                        default false;
                        description "This attribute represents the default value for enabling
Subscriber Management filters and controls in the CM if the parameter is not signaled in the DOCSIS
Registration process.";
                    }
                    leaf cpe-learnable {
                        type boolean;
                        default true;
                        description "This attribute represents the default value for enabling the
CPE learning process for the CM if the parameter is not signaled in the DOCSIS Registration
process.";
                    }
                    leaf subscriber-downstream-filter {
                        type uint16 {
                            range "0..1024";
                        }
                        default 0;
                        description "This attribute represents the default value for the subscriber
(CPE) downstream filter group for the CM if the parameter is not signaled in the DOCSIS Registration
process.";
                    }
                    leaf subscriber-upstream-filter {
                        type uint16 {
                            range "0..1024";
                        }
                        default 0;
                        description "This attribute represents the default value for the subscriber
(CPE) upstream filter group for the CM if the parameter is not signaled in the DOCSIS Registration
process.";
                    }
                    leaf cm-downstream-filter {
                        type uint16 {
                            range "0..1024";
                        }
                        default 0;
                        description "This attribute represents the default value for the CM stack
downstream filter group applying to the CM if the parameter is not signaled in the DOCSIS
Registration process.";
                    }
                    leaf cm-upstream-filter {
                        type uint16 {
                            range "0..1024";
                        }
                        default 0;
```

```
                        description "This attribute represents the default value for the CM stack
upstream filter group for the CM if the parameter is not signaled in the DOCSIS Registration
process.";
                    }
                    leaf ps-downstream-filter {
                        type uint16 {
                            range "0..1024";
                        }
                        default 0;
                        description "This attribute represents the default value for the PS or
eRouter downstream filter group for the CM if the parameter is not signaled in the DOCSIS
Registration process.";
                    }
                    leaf ps-upstream-filter {
                        type uint16 {
                            range "0..1024";
                        }
                        default 0;
                        description "This attribute represents the default value for the PS or
eRouter upstream filter group for the CM if the parameter is not signaled in the DOCSIS Registration
process.";
                    }
                    leaf mta-downstream-filter {
                        type uint16 {
                            range "0..1024";
                        }
                        default 0;
                        description "This attribute represents the default value for the MTA
downstream filter group for the CM if the parameter is not signaled in the DOCSIS Registration
process.";
                    }
                    leaf mta-upstream-filter {
                        type uint16 {
                            range "0..1024";
                        }
                        default 0;
                        description "This attribute represents the default value for the MTA
upstream filter group for the CM if the parameter is not signaled in the DOCSIS Registration
process.";
                    }
                    leaf stb-downstream-filter {
                        type uint16 {
                            range "0..1024";
                        }
                        default 0;
                        description "This attribute represents the default value for the STB
downstream filter group for the CM if the parameter is not signaled in the DOCSIS Registration
process.";
                    }
                    leaf stb-upstream-filter {
                        type uint16 {
                            range "0..1024";
                        }
                        default 0;
                        description "This attribute represents the default value for the STB
upstream filter group for the CM if the parameter is not signaled in the DOCSIS Registration
process.";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list filter-group {
                    key "rule-id group-id";
                    min-elements 0;
                    description
                        "This object describes a set of filter or classifier criteria. Classifiers
are assigned by group to the individual CMs. That assignment is made via the Subscriber Management
TLVs encodings sent upstream from the CM to the CMPA during registration, or in their absence,
default values configured in the CCAP.
                         A Filter Group ID (GrpId) is a set of rules that correspond to the
expansion of a UDC Group ID into individual UDC rules. The UDC Group IDs are linked to Ids of the
FilterGrp object so the CCAP can signal those filter rules as UDCs to the CM during the registration
process. Implementation of L2 classification criteria is optional for the CCAP; LLC/MAC upstream and
downstream filter criteria can be ignored during the packet matching process.";
                    reference
                        "DOCSIS 3.0 Operations Support System Interface Specification
```

```
                        CM-SP-OSSIv3.0-I15-110623 FilterGrp Object section.";
                    leaf group-id {
                        type uint16 {
                            range "1..1024";
                        }
                        mandatory true;
                        description "This key is an identifier for a set of classifiers known as a
filter group. Each CM may be associated with several filter groups for its upstream and downstream
traffic, one group per target end point on the CM as defined in the Grp object. Typically, many CMs
share a common set of filter groups. The range for this attribute is 1 to 1024 to align it with the
values used in the Base Object.";
                    }
                    leaf rule-id {
                        type uint16 {
                            range "1..65535";
                        }
                        mandatory true;
                        description "This key represents an ordered classifier identifier within the
group. Filters are applied in order if the Priority attribute is not supported.";
                    }
                    leaf filter-action {
                type filter-action-type;
                        default permit;
                        description "This attribute represents the action to take upon this filter
matching.";
                    }
                    leaf priority {
                        type uint16;
                        default 0;
                        description "This attribute defines the order in which the classifiers are
compared against packets. The higher the value, the higher the priority.";
                    }
                    leaf ip-tos-low {
                        type ccap-octet-data-type {
                            length "2";
                        }
                        default 00;
                        description "This attribute represents the low value of a range of ToS (Type
of Service) octet values. The IP ToS octet, as originally defined in [RFC 791], has been superseded
by the 6-bit Differentiated Services Field (DSField, [i.10]) and the 2-bit Explicit Congestion
Notification Field (ECN field, [i.9]). This attribute is defined as an 8-bit octet as per the DOCSIS
Specification for packet classification.
";
                        reference
                            "DOCSIS 3.0 MAC and Upper Layer Protocols Interface Specification
                             CM-SP-MULPIv3.0-I16-110623; RFC 791; RFC 3168; RFC 3260.";
                    }
                    leaf ip-tos-high {
                        type ccap-octet-data-type {
                            length "2";
                        }
                        default 00;
                        description "This attribute represents the high value of a range of ToS
octet values. The IP ToS octet, as originally defined in [RFC 791], has been superseded by the 6-bit
Differentiated Services Field (DSField, [i.10]) and the 2-bit Explicit Congestion Notification Field
(ECN field, [i.9]). This attribute is defined as an 8-bit octet as per the DOCSIS Specification for
packet classification.
";
                        reference
                            "DOCSIS 3.0 MAC and Upper Layer Protocols Interface Specification
                             CM-SP-MULPIv3.0-I16-110623; RFC 791; RFC 3168; RFC 3260.";
                    }
                    leaf ip-tos-mask {
                        type ccap-octet-data-type {
                            length "2";
                        }
                        default 00;
                        description "This attribute represents the mask value that is bitwise ANDed
with ToS octet in an IP packet, and the resulting value is used for range checking of IpTosLow and
IpTosHigh.";
                    }
                    leaf ip-protocol {
                        type uint16 {
                            range "0..257";
                        }
                        default 256;
                        description "This attribute represents the value of the IP Protocol field
required for IP packets to match this rule. The value 256 matches traffic with any IP Protocol
```

```
value. The value 257 by convention matches both TCP and UDP.";


                    }
                    leaf source-address {
                        type inet:ip-prefix;
                        mandatory true;
                        description "This attribute specifies the value of the IP Source Address
required for packets to match this rule and which bits of a packet's IP Source Address are compared
to match this rule. An IP packet matches the rule when the packet's IP Source Address bitwise ANDed
with the mask value defined by the prefix equals the InetSrcAddr value. ";
                    }
                    leaf destination-address {
                        type inet:ip-prefix;
                        mandatory true;
                        description "This attribute specifies the value of the IP Destination
Address required for packets to match this rule and which bits of a packet's IP Source Address are
compared to match this rule. An IP packet matches the rule when the packet's IP Destination Address
bitwise ANDed with the mask value defined by the prefix equals the InetDestAddr value.";
                    }
                    leaf source-port-start {
                        type inet:port-number;
                        default 0;
                        description "This attribute represents the low-end inclusive range of
TCP/UDP source port numbers to which a packet is compared. This attribute is irrelevant for non-
TCP/UDP IP packets.";
                    }
                    leaf source-port-end {
                        type inet:port-number;
                        default 65535;
                        description "This attribute represents the high-end inclusive range of
TCP/UDP source port numbers to which a packet is compared. This attribute is irrelevant for non-
TCP/UDP IP packets.";
                    }
                    leaf destination-port-start {
                        type inet:port-number;
                        default 0;
                        description "This attribute represents the low-end inclusive range of
TCP/UDP destination port numbers to which a packet is compared. This attribute is irrelevant for
non-TCP/UDP IP packets.";
                    }
                    leaf destination-port-end {
                        type inet:port-number;
                        default 65535;
                        description "This attribute represents the high-end inclusive range of
TCP/UDP destination port numbers to which a packet is compared. This attribute is irrelevant for
non-TCP/UDP IP packets.";
                    }
                    leaf destination-mac-address {
                        type yang:mac-address;
                        default 00:00:00:00:00:00;
                        description "This attribute represents the criteria to match against an
Ethernet frame MAC address bitwise ANDed with DestMacMask.";
                    }
                    leaf destination-mac-mask {
                        type yang:mac-address;
                        default 00:00:00:00:00:00;
                        description "An Ethernet frame matches an entry when its destination MAC
address bitwise ANDed with the DestMacMask attribute equals the value of the DestMacAddr
attribute.";
                    }
                    leaf source-mac-address {
                        type yang:mac-address;
                        default FF:FF:FF:FF:FF:FF;
                        description "This attribute represents the value to match against an
Ethernet frame source MAC address.";
                    }
                    leaf ethernet-protocol-id {
                type ethernet-protocol-id-type;
                        default none;
                        description
                            "This attribute indicates the format of the layer 3 protocol ID in the
Ethernet frame.";
                    }
                    leaf ethernet-protocol {
                        type uint16;
                        default 0;
```

```
                        description "This attribute represents the Ethernet protocol type to be
matched against the frames. For EnetProtocolType set to 'none', this attribute is ignored when
considering whether a packet matches the current rule. If the attribute EnetProtocolType is
'ethertype', this attribute gives the 16-bit value of the EtherType that the packet needs to match
in order to match the rule. If the attribute EnetProtocolType is 'dsap', the lower 8 bits of this
attribute's value needs to match the DSAP byte of the packet in order to match the rule. If the
Ethernet frame contains an 802.1p/Q Tag header (i.e. EtherType 0x8100), this attribute applies to
the embedded EtherType field within the 802.1p/Q header.";
                    }
                    leaf user-priority-low {
                        type uint8 {
                            range "0..7";
                        }
                        default 0;
                        description "This attribute applies only to Ethernet frames using the
802.1p/Q tag header (indicated with EtherType 0x8100). Such frames include a 16-bit Tag that
contains a 3-bit Priority field and a 12-bit VLAN number. Tagged Ethernet frames needs to have a 3-
bit Priority field within the range of PriLow to PriHigh in order to match this rule.";
                    }
                    leaf user-priority-high {
                        type uint8 {
                            range "0..7";
                        }
                        default 7;
                        description "This attribute applies only to Ethernet frames using the
802.1p/Q tag header (indicated with EtherType 0x8100). Such frames include a 16-bit Tag that
contains a 3-bit Priority field and a 12-bit VLAN number. Tagged Ethernet frames need to have a 3-
bit Priority field within the range of PriLow to PriHigh in order to match this rule.";
                    }
                    leaf vlan-id {
                        type uint16 {
                            range "0 | 1..4094";
                        }
                        default 0;
                        description "This attribute applies only to Ethernet frames using the
802.1p/Q tag header. Tagged packets need to have a VLAN Identifier that matches the value in order
to match the rule.";
                    }
                    leaf flow-label {
                        type uint32 {
                            range "0..1048575";
                        }
                        default 0;
                        description
                            "This attribute represents the Flow Label field in the IPv6 header to be
matched by the classifier.
                             The value zero indicates that the Flow Label is not specified as part
of the classifier and is not matched against packets.";
                    }
                    leaf cm-interface-mask {
                        type bits {
                            bit eCm {
                                position 0;
                            }
                            bit cmci {
                                position 1;
                            }
                            bit docsCableMacLayer {
                                position 2;
                            }
                            bit docsCableDownstream {
                                position 3;
                            }
                            bit docsCableUpstream {
                                position 4;
                            }
                            bit unused-5 {
                                position 5;
                            }
                            bit unused-6 {
                                position 6;
                            }
                            bit unused-7 {
                                position 7;
                            }
                            bit unused-8 {
                                position 8;
                            }
```

```
                            bit unused-9 {
                                position 9;
                            }
                            bit unused-10 {
                                position 10;
                            }
                            bit unused-11 {
                                position 11;
                            }
                            bit unused-12 {
                                position 12;
                            }
                            bit unused-13 {
                                position 13;
                            }
                            bit unused-14 {
                                position 14;
                            }
                            bit unused-15 {
                                position 15;
                            }
                            bit eMta {
                                position 16;
                            }
                            bit eStbIp {
                                position 17;
                            }
                            bit eStbDsg {
                                position 18;
                            }
                        }
                        description "This attribute represents a bit-mask of the CM in-bound
interfaces to which this classifier applies. This attribute only applies to upstream Drop
Classifiers being sent to CMs during the registration process.";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            container docs-qos {
                description "This group of configuration elements allows for the configuration of
DOCSIS QoS.";
                list service-class {
                    key service-class-name;
                    min-elements 0;
                    description "This object describes a provisioned service class on a CCAP. Each
object instance defines a template for certain DOCSIS QOS Parameter Set values. When a CM creates or
modifies an Admitted QOS Parameter Set for a Service Flow, it may reference a Service Class Name
instead of providing explicit QOS Parameter Set values. In this case, the CCAP populates the QOS
Parameter Set with the applicable corresponding values from the named Service Class. Subsequent
changes to a Service Class row do not affect the QOS Parameter Set values of any service flows
already admitted. A service class template applies to only a single direction, as indicated in the
ServiceClassDirection attribute.";
                    reference
                        "DOCSIS 3.0 Operations Support System Interface Specification
                         CM-SP-OSSIv3.0-I15-110623 ServiceClass Object section.";
                    leaf service-class-name {
                        type string {
                            length "1..15";
                        }
                        description "This key indicates the Service Class Name associated with this
object instance. DOCSIS specifies that the maximum size is 16 ASCII characters including a
terminating zero. The terminating zero is not represented in this SnmpAdminString syntax attribute.
";
                        reference
                            "DOCSIS 3.0 MAC and Upper Layer Protocols Interface Specification
                             CM-SP-MULPIv3.0-I16-110623 Service Class Name section in the Common
Radio Frequency Interface Encodings Annex";
                    }
                    leaf priority {
                        type uint8;
                        default 0;
```

```
                        description "This attribute is the template for the Priority attribute of
the QoS Parameter Set.";
                    }
                    leaf max-traffic-rate {
                        type uint32 {
                            range "0..4294967295";
                        }
                        units bps;
                        default 0;
                        description "This attribute is the template for the MaxTrafficRate attribute
of the QoS Parameter Set.";
                    }
                    leaf max-traffic-burst {
                        type uint32;
                        units bytes;
                        default 3044;
                        description "This attribute is the template for the MaxTrafficBurst
attribute of the QoS Parameter Set.";
                    }
                    leaf min-reserved-rate {
                        type uint32 {
                            range "0..4294967295";
                        }
                        units bps;
                        default 0;
                        description "This attribute is the template for the MinReservedRate
attribute of the QoS Parameter Set.";
                    }
                    leaf min-reserved-packet {
                        type uint16;
                        units bytes;
                        mandatory true;
                        description "This attribute is the template for the MinReservedPkt attribute
of the QoS Parameter Set.";
                    }
                    leaf max-concatenated-burst {
                        type uint16;
                        units bytes;
                        default 1522;
                        description "This attribute is the template for the MaxConcatBurst attribute
of the QoS Parameter Set.";
                    }
                    leaf nominal-polling-interval {
                        type uint32;
                        units microseconds;
                        default 0;
                        description "This attribute is the template for the NomPollInterval
attribute of the QoS Parameter Set.";
                    }
                    leaf tolerated-poll-jitter {
                        type uint32;
                        units microseconds;
                        default 0;
                        description "This attribute is the template for the TolPolJitter attribute
of the QoS Parameter Set.";
                    }
                    leaf unsolicited-grant-size {
                        type uint16;
                        units bytes;
                        default 0;
                        description "This attribute is the template for the UnsolicitGrantSize
attribute of the QoS Parameter Set.";
                    }
                    leaf nominal-grant-interval {
                        type uint32;
                        units bytes;
                        default 0;
                        description "This attribute is the template for the NomGrantInterval
attribute of the QoS Parameter Set.";
                    }
                    leaf tolerated-grant-jitter {
                        type uint32;
                        units microseconds;
                        default 0;
                        description "This attribute is the template for the TolGrantJitter attribute
of the QoS Parameter Set.";
                    }
                    leaf grants-per-interval {
```

```
                        type uint8;
                        units dataGrants;
                        default 0;
                        description "This attribute is the template for the GrantsPerInterval
attribute of the QoS Parameter Set.";
                    }
                    leaf max-latency {
                        type uint32;
                        units microseconds;
                        default 0;
                        description "This attribute is the template for the MaxLatency attribute of
the QoS Parameter Set.";
                    }
                    leaf active-timeout {
                        type uint16;
                        units seconds;
                        default 0;
                        description "This attribute is the template for the ActiveTimeout attribute
of the QoS Parameter Set.";
                    }
                    leaf admitted-timeout {
                        type uint16;
                        units seconds;
                        default 200;
                        description "This attribute is the template for the AddmitedTimeout
attribute of the QoS Parameter Set.";
                    }
                    leaf scheduling-type {
                type service-flow-scheduling-type;
                        default best-effort;
                        description "This attribute is the template for the SchedulingType attribute
of the QoS Parameter Set.";
                    }
                    leaf request-policy {
                        type ccap-octet-data-type {
                            length "8";
                        }
                        default 00000000;
                        description "This attribute is the template for the RequestPolicyOct
attribute of the QoS Parameter Set.";
                    }
                    leaf tos-and-mask {
                        type ccap-octet-data-type {
                            length "2";
                        }
                        mandatory true;
                        description "This attribute is the template for the TosAndMask attribute of
the QoS Parameter Set.";
                    }
                    leaf tos-or-mask {
                        type ccap-octet-data-type {
                            length "2";
                        }
                        mandatory true;
                        description "This attribute is the template for the TosOrMask attribute of
the QoS Parameter Set.";
                    }
                    leaf direction {
                type direction-type;
                        default upstream;
                        description "This attribute is the template for the Direction attribute of
the QoS Parameter Set.";
                    }
                    leaf dscp-overwrite {
                        type int32 {
                            range "-1 | 0..63";
                        }
                        default -1;
                        description
                            "This attribute allows the overwrite of the DSCP field per RFC 3260.
                             If this attribute is -1, then the corresponding TosAndMask value is set
to be 'FF'H and TosOrMask is set to '00'H. Otherwise, this attribute is in the range of 0..63, and
the corresponding TosAndMask value is '03'H and TosOrMaskvalue is this attribute value shifted left
by two bit positions.";
                    }
                    leaf required-attribute-mask {
                        type attribute-mask-type;
```

```
                        description "This attribute is the template for the RequiredAttrMask
attribute of the QoS Parameter Set.";
                    }
                    leaf forbidden-attribute-mask {
                        type attribute-mask-type;
                        description "This attribute is the template for the ForbiddenAttrMask
attribute of the QoS Parameter Set.";
                    }
                    leaf attribute-aggregate-rule-mask {
                        type ccap-octet-data-type {
                            length "8";
                        }
                        default 00000000;
                        description "This attribute is the template for the AttrAggregationMask
attribute of the QoS Parameter Set.";
                    }
                    leaf application-id {
                        type uint32;
                        mandatory true;
                        description "This attribute is the template for the AppId attribute of the
QoS Parameter Set.";
                    }
                    leaf multiplier-contention-request-window {
                        type uint8 {
                            range "4..12";
                        }
                        units eighths;
                        default 8;
                        description "This attribute is the template for the
MultiplierContentionReqWindow attribute of the QoS Parameter Set.";
                    }
                    leaf multiplier-bytes-requested {
                        type uint8 {
                            range "1 | 2 | 4 | 8 | 16";
                        }
                        default 4;
                        description "This attribute is the template for the MultiplierBytesReq
attribute of the QoS Parameter Set.";
                    }
                    leaf max-requests-per-sid-cluster {
                        type uint8;
                        units requests;
                        default 0;
                        description
                            "This attribute is the template for the MaxReqPerSidCluster attribute of
the QoS Parameter Set.
                             This attribute has been deprecated and replaced with
MaxReqPerSidCluster in the ServiceFlow object.";
                    }
                    leaf max-outstanding-bytes-per-sid-cluster {
                        type uint32;
                        units bytes;
                        default 0;
                        description
                            "This attribute is the template for the MaxOutstandingBytesPerSidCluster
attribute of the QoS Parameter Set.
                             This attribute has been deprecated and replaced with
MaxOutstandingBytesPerSidCluster in the ServiceFlow object.";
                    }
                    leaf max-total-bytes-requested-per-sid-cluster {
                        type uint32;
                        units bytes;
                        default 0;
                        description
                            "This attribute is the template for the MaxTotBytesReqPerSidCluster
attribute of the QoS Parameter Set.
                             This attribute has been deprecated and replaced with
MaxTotBytesReqPerSidCluster in the ServiceFlow object.";
                    }
                    leaf max-time-in-sid-cluster {
                        type uint16;
                        units milliseconds;
                        default 0;
                        description
                            "This attribute is the template for the MaxTimeInSidCluster attribute of
the QoS Parameter Set.
                             This attribute has been deprecated and replaced with
MaxTimeInSidCluster in the ServiceFlow object.";
```

```
                    }
                    leaf peak-traffic-rate {
                        type uint32;
                        units bps;
                        default 0;
                        description "This attribute is the template for the PeakTrafficRate
attribute of the QoS Parameter Set.";
                    }
                    leaf ds-resequencing {
                type ds-resequencing-type;
                        default resequencing-dsid;
                        description "This attribute is the template for the DsResequencing attribute
of the QoS Parameter Set.";
                    }
                    leaf minimum-buffer {
                        type uint32;
                        units bytes;
                        default 0;
                        description "This attribute is the template for the MinimumBuffer attribute
of the QoS Parameter Set.";
                    }
                    leaf target-buffer {
                        type uint32;
                        units bytes;
                        default 0;
                        description "This attribute is the template for the TargetBuffer attribute
of the QoS Parameter Set.";
                    }
                    leaf maximum-buffer {
                        type uint32;
                        units bytes;
                        default 4294967295;
                        description "This attribute is the template for the MaximumBuffer attribute
of the QoS Parameter Set.";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list qos-profile {
                    key qos-profile-index;
                    description "The qos-profile object is used to help provide a mapping between
Cable Modems that have registered with a DOCSIS 1.0 style Class of Service. The support for this
configuration is dependent on the CCAP supporting DOCSIS 1.0 style configuration files and CM
registrations. ";
                    reference "RFC 4546 docsIfQosProfileTable";
                    leaf qos-profile-index {
                        type uint16 {
                            range "1..16383";
                        }
                        description "This attribute configures an unique index for each instance of
this object.";
                    }
                    leaf priority {
                        type uint8 {
                            range "0..7";
                        }
                        default 0;
                        description "This attribute configures the DOCSIS priority for this service
flow.";
                    }
                    leaf max-up-bandwidth {
                        type uint32 {
                            range "0..100000000";
                        }
                        units bps;
                        default 0;
                        description "This attribute configures the maximum upstream bandwidth, in
bits per second, allowed for a service with this service class. The value zero is used if there is
no restriction of upstream bandwidth.";
                    }
                    leaf guaranteed-up-bandwidth {
                        type uint32 {
                            range "0..100000000";
                        }
                        units bps;
                        default 0;
```

```
                        description "This attribute configures the minimum guaranteed upstream
bandwidth, in bits per second allowed for a service with this service class.";
                    }
                    leaf max-down-bandwidth {
                        type uint32 {
                            range "0..100000000";
                        }
                        units bps;
                        default 0;
                        description "This attribute configures the maximum downstream bandwidth, in
bits per second allowed for a service with this service class. The value of zero is used if there is
no restriction of downstream bandwidth.";
                    }
                    leaf baseline-privacy {
                        type boolean;
                        default false;
                        description "This attribute configures BPI encryption for this service
class.";
                    }
                    leaf max-transmit-burst {
                        type uint16;
                        units bytes;
                        default 0;
                        description "This attribute configures the maximum number of bytes that may
be requested for a single upstream transmission.  A value of zero means there is no limit.  Note:
This value does not include any physical layer overhead.";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            container docs-multicast-qos {
                description "This group of configuration elements allows for the configuration of
DOCSIS Multicast QoS. They are based on the configuration elements from [OSSIv3.0].";
                leaf default-group-service-class {
                    type leafref {
                        path "../../docs-qos/service-class/service-class-name";
                    }
                    description "This object provides the name of the Default Group Service Class.
The CCAP instantiates a Default Group Service Flow with the QOS param Set indicated by this Service
Class Name reference on every Downstream Channel Set to which it replicates multicast packets that
are otherwise unclassified by a Group Classifier Rule.
Reference: [OSSIv3.0], DefGrpSvcClass Object section";
                }
                list group-phs-config {
                    key group-phs-config-id;
                    min-elements 0;
                    description
                        "This object controls the configuration of DSID-indexed PHS for multicast
sessions. Configuration of PHS Rules via this object are applied to individual multicast sessions
even if the referenced GrpCfg object identified a GrpQosCfg instance with a QosCtrl of
'aggregateSession'.
                         This object supports the creation and deletion of instances.
                         Creation of multiple instances of this object require the following
attributes to be set:
                         - PhsField
                         - PhsMask
                         - PhsSize";
                    reference
                        "DOCSIS 3.0 Operations Support System Interface Specification
                         CM-SP-OSSIv3.0-I15-110623 CmtsGrpPhsCfg Object section.";
                    leaf group-phs-config-id {
                        type uint16 {
                            range "1..65535";
                        }
                        mandatory true;
                        description "This attribute identifies the unique identifier of a PHS rule
that is referenced by the GrpCfg object.";
                    }
                    leaf phs-field {
                        type ccap-octet-data-type {
                            length "0..510";
```

```
                    }
                    mandatory true;
                    description "This attribute defines the bytes of the DOCSIS header which are
to be suppressed/restored by the sending/receiving device.";
                }
                leaf phs-mask {
                    type ccap-octet-data-type {
                        length "0..64";
                    }
                    mandatory true;
                    description
                        "This attribute defines the bit mask which is used in combination with
the PhsField to define which bytes in header need to be suppressed/restored by the sending or
receiving device.
                         Each bit of this bit mask corresponds to a byte in the PhsField, with
the least significant bit corresponding to the first byte of the PhsField.
                         Each bit of the bit mask specifies whether or not the corresponding
byte should be suppressed in the packet. A bit value of '1' indicates that the byte should be
suppressed by the sending device and restored by the receiving device.
                         A bit value of '0' indicates that the byte should not be suppressed by
the sending device or restored by the receiving device.
                         If the bit mask does not contain a bit for each byte in the PhsField
then the bit mask is extended with bit values of '1' to be the necessary length.";
                }
                leaf phs-size {
                    type uint8;
                    units bytes;
                    mandatory true;
                    description
                        "This attribute specifies the number of bytes in the header to be
suppressed and restored.
                         The value of this object matches the number of bytes the bits indicated
in the PhsField attribute.";
                }
                leaf phs-verify {
                    type boolean;
                    default false;
                    description "If this attribute specifies the Payload Header Suppression
verification value of 'true' the sender shall verify PhsField is the same as what is contained in
the packet to be suppressed.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list group-config {
                key group-config-id;
                min-elements 0;
                description
                    "This object controls the QoS, PHS, and encryption settings for downstream
forwarding of IP multicast sessions. An IP multicast session is replicated to one or more Downstream
Channel Sets (DCSs), where each DCS is either a single downstream channel or a downstream bonding
group of multiple channels. The CCAP determines on which DCSs to replicate a multicast session based
on IP multicast membership reports (joins) or other vendor-specific static configuration.
                     The CmtsGrpCfg object allows for the configuration of a range of sessions
through the SrcPrefixAddr and GrpPrefixAddr and SrcPrefixLen and GrpPrefixLen attributes.
                     Cable operators can specify configuration rules for a range of multicast
sessions through the tuples of (SrcPrefixAddr, SrcPrefixLen, GrpPrefixAddr, GrpPrefixLen) attributes
in an entry. The QosCfgId attribute identifies the QoS rule, the EncryptCfgId identifies the
encryption rule and the PhsCfgId identifies the PHS rule for a particular entry. Even if an entry
indicates a range of multicast sessions, the Encryption and PHS rules are applied on a per-session
basis. Thus, when an Operator configures PHS rules or Encryption for a given GroupConfig entry, each
session has those rules applied on a per session and per replication basis. Group PHS and Group
Encryption rules are indicated by using a non-zero value for the PhsCfgId and EncryptCfgId,
respectively.
                     The DocsMcastCmtsGrpQosCfgQosCtrl attribute from the CmtsGrpQosCfg object
is used to determine if the traffic for a range of multicast sessions identified by an entry in the
CmtsGrpCfg object will be transmitted in an Aggregate-Session Group Service Flow or will be
transmitted separately for each session using Single-Session Group Service Flows. Even if the range
of multicast sessions are transmitted on an Aggregate-Session Group Service Flow, the PHS and
Encryption rules are always applied individually to a multicast session on a per-session DSID basis
prior to being transmitted on an Aggregate-Session Group Service Flow (GSF).
                     Creation of a new instance of this object requires the following attributes
to be set.
                     - RulePriority
                     - SrcPrefixAddr
                     - GrpPrefixAddr
```

```
                        - TosLow
                        - TosHigh
                        - TosMask";
                    reference
                        "DOCSIS 3.0 Operations Support System Interface Specification
                         CM-SP-OSSIv3.0-I15-110623 CmtsGrpCfg Object section.";
                    leaf group-config-id {
                        type uint32 {
                            range "1..4294967295";
                        }
                        description "This attribute represents the unique identifier of instances of
this object. This attribute is the key that identifies unique instances of the group-config
Object.";
                    }
                    leaf rule-priority {
                        type uint8;
                        mandatory true;
                        description "This attribute indicates the priority of this entry used to
resolve which instance of this object apply when a newly replicated multicast session matches
multiple entries. Higher values indicate a higher priority. Valid values for this attribute are
0..63 and 192..255 in order to not conflict with CMTS internally-created instances that use the
range 64..191.";
                    }
                    uses ssm-group;
                    leaf tos-low {
                        type ccap-octet-data-type {
                            length "2";
                        }
                        mandatory true;
                        description
                            "This attribute identifies the low value of a range of the TOS byte
value to be defined in a packet classifier this GC instantiates in the GCR in order to limit the
GCR-matched traffic to a particular set of DSCPs. This applies to the IPv4 TOS byte and the IPv6
Traffic Class byte.
                             The IP TOS octet, as originally defined in [RFC 791], has been
superseded by the 6-bit Differentiated Services Field and the 2-bit Explicit Congestion Notification
Field.";
                        reference "RFC 791; RFC 3260; RFC 3168.";
                    }
                    leaf tos-high {
                        type ccap-octet-data-type {
                            length "2";
                        }
                        mandatory true;
                        description
                            "This attribute identifies the high value of a range of the TOS byte
value to be defined in a packet classifier this GC instantiates in the GCR in order to limit the
GCR-matched traffic to a particular set of DSCPs. This applies to the IPv4 TOS byte and the IPv6
Traffic Class byte.
                             The IP TOS octet, as originally defined in [RFC 791], has been
superseded by the 6-bit Differentiated Services Field (DSField, [i.10]) and the 2-bit Explicit
Congestion Notification Field (ECN field, [i.9]).";
                        reference "RFC 791; RFC 3260; RFC 3168.";
                    }
                    leaf tos-mask {
                        type ccap-octet-data-type {
                            length "2";
                        }
                        mandatory true;
                        description
                            "This attribute identifies the mask value bitwise ANDed with a TOS byte
value to be defined in a packet classifier this GC instantiates in the GCR in order to limit the
GCR-matched traffic to a particular set of DSCPs. This applies to the IPv4 TOS byte and the IPv6
Traffic Class byte.
                             The IP TOS octet, as originally defined in [RFC 791], has been
superseded by the 6-bit Differentiated Services Field (DSField, [i.10]) and the 2-bit Explicit
Congestion Notification Field (ECN field, [i.9]).";
                        reference "RFC 791; RFC 3260; RFC 3168.";
                    }
                    leaf group-qos-config-id {
                        type leafref {
                            path "../../group-qos-config/group-qos-config-id";
                        }
                        description "This attribute identifies an instance in CmtsGrpQosCfg for
configuring the QoS for the replication of the sessions matching this CmtsGrpCfg instance.
";
                    }
                    leaf group-encryption-config-id {
```

```
                        type leafref {
                            path "../../group-encryption-config/group-encryption-config-id";
                        }
                        description "This attribute identifies an instance in CmtsGrpEncryptCfg for
configuring the encryption of replications derived from this GC.";
                    }
                    leaf group-phs-config-id {
                        type leafref {
                            path "../../group-phs-config/group-phs-config-id";
                        }
                        description "This attribute identifies an instance in CmtsGrpPhsCfg that
configures DSID-indexed PHS compression for all replications derived from this GC.";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list group-encryption-config {
                    key group-encryption-config-id;
                    description "This object controls the configuration of the Security Association
(SA) and the encryption algorithm used for multicast sessions.

";
                    reference
                        "DOCSIS 3.0 Operations Support System Interface Specification
                         CM-SP-OSSIv3.0-I15-110623 CmtsGrpEncryptCfg Object section.";
                    leaf group-encryption-config-id {
                        type uint16;
                        description "This attribute specifies the unique identifier of instances of
this object.";
                    }
                    leaf control {
                type encryption-algorithm-ctl-type;
                        default mgmt;
                        description "This attribute controls whether the CMTS can select the
encryption algorithm or if this can be set manually using the Alg attribute.";
                    }
                    leaf algorithm {
                type encryption-algorithm-type;
                        default des56-cbc-mode;
                        description "This attribute defines which encryption algorithm will be used
for an SA referenced by this object when the Ctrl is set to 'mgmt'.";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list group-qos-config {
                    key group-qos-config-id;
                    min-elements 0;
                    description
                        "This object configures the QoS configured for Multicast sessions replicated
to any Downstream Channel Set. It does not control which particular DCSs to which the CCAP
replicates a multicast session.
                         An instance of this object is called a GQC entry. A GQC entry controls how
the CCAP instantiates a Group Classifier Rule (GCR) on the DCS to match packets of the multicast
session. A GCR uses source and destination IP address and ToS criteria.
                         A GQC entry controls how and with what QoS parameters a GSF is created on a
DCS. All downstream multicast packets are scheduled on a GSF. The QoS Type attribute of the GQC
entry controls whether the CCAP creates one GSF for each single IP multicast session or whether the
CCAP creates one GSF for the aggregate of all sessions that match the GQC criteria. The GQC instance
contains a reference to a Service Class Name QoS Parameter Set template. The Service Class defines
the list of QoS parameters for the GSF(s) instantiated for the GQC entry.
                         A CCAP identifies one Service Class as the Default Group QoS Service Class.
The CCAP instantiates a Default Group Service Flow on each single-channel DCS based on the
parameters of the Default Group QoS Service Class.
                         The set of GCRs and GSFs instantiated on a DCS control how QoS is provided
to multicast packets replicated to the DCS. For each multicast packet, the CCAP classifies the
packet to the highest priority matching GCR on that DCS. The GCR refers to a single GSF, which
controls the scheduling of the packets on the DCS. If the multicast packet does not match any GCR on
the DCS, the packet is scheduled on the Default Group Service Flow of the DCS. The CCAP replicates
unclassified multicast traffic to only DCSs consisting of a single downstream channel. Thus, the
Maximum Sustained Traffic Rate QoS parameter of the Default Group Service Class limits the aggregate
rate of unclassified multicast traffic on each downstream channel.
                         The CCAP is expected to instantiate GCRs and GSFs controlled by the entries
in this table only for the duration of replication of the multicast sessions matching the entry.
```

```
                        This object supports the creation of multiple instances.
                        Creation of new instances of this object require the following objects to
be set:
                        - SvcClassName
                        - QosCtrl
                        - AggSessLimit";
                    reference
                        "DOCSIS 3.0 Operations Support System Interface Specification
                        CM-SP-OSSIv3.0-I15-110623 CmtsGrpQosCfg Object section.";
                    leaf group-qos-config-id {
                        type uint16;
                        mandatory true;
                        description "This attribute identifies a unique Group QoS Configuration
object instance.";
                    }
                    leaf service-class-name {
                        type leafref {
                            path "/ccap/docsis/docs-qos/service-class/service-class-name";
                        }
                        mandatory true;
                        description "Reference to the Service Class Name.";
                    }
                    leaf qos-control {
                        type qos-control-type;
                        mandatory true;
                        description "This attribute identifies how Group Classifier Rules (GCRs) and
Group Service Flows (GSFs) are instantiated when multiple sessions match the (S,G) criteria of this
entry.";
                    }
                    leaf aggregated-session-limit {
                        type uint16 {
                            range "1.. 65535";
                        }
                        units sessions;
                        mandatory true;
                        description "This attribute identifies the maximum number of sessions that
may be aggregated in an aggregated Service Flow. This value is ignored in case of a GQC entry with
QosCtrl set to 'singleSession'.";
                    }
                    leaf application-id {
                        type uint32;
                        default 0;
                        description
                            "This attribute allows the operator to configure a Cable Operator
defined Application Identifier for multicast sessions, e.g. an Application Manager ID and
Application Type. This Application Identifier can be used to influence admission control or other
policies in the CMTS that are outside of the scope of the present document. This parameter is
optional in defining QoS for multicast sessions.
                            If the value of this attribute is different from the value of the AppId
in the referenced SCN for this GQC instance, the value of this attribute is used.";
                        reference
                            "DOCSIS 3.0 MAC and Upper Layer Protocols Interface Specification
                            CM-SP-MULPIv3.0-I16-110623 Application Identifier section in the Common
Radio Frequency Interface Encodings Annex
                            PacketCable Multimedia Specification PKT-SP-MM-I05-091029 Policy Server
and CMTS Interface section.";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            choice remote-query {
                case cm-remote-query {
                    leaf enable {
                        type boolean;
                        default false;
                    }
                    leaf snmp-community {
                        type string {
                            length "1..32";
                        }
                        mandatory true;
```

```
                    }
                    leaf polling-interval {
                        type uint32;
                        units "seconds";
                        description "This attribute configures the minimum amount of time in seconds
                            between consecutive polls of the same MIB object on the same cable
modem.";
                    }
                    leaf source-ip-interface-name {
                        type string;
                        description "Local IP interface; needs to be a configured ip-interface name.
If not specified, then the vendor picks the source IP address.
                             The source interface address used, when configured, is the configured
PrimaryIpv4 or (one of) Global Scope Ipv6 address for the specified interface.";
                    }
                    container yang-ext4 {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
            }
            container docs-mac-domain {
                description "This container contains MAC domain level control and configuration.";
                list downstream-bonding-group {
                    key bonding-group-name;
                    min-elements 0;
                    description "The downstream-bonding-group object allows for the static creation
of Downstream bonding groups. In some current DOCSIS 3.0 configurations, downstream channels are not
tied directly to a specific MAC domain, while in others these downstream channels are an integral
part of the MAC domain. For CCAP flexibility, downstream channels that are configured into a bonding
group will not be directly associated with a specific MAC domain.";
                    leaf bonding-group-name {
                        type string;
                        description "The name of the bonding group.";
                    }
                    leaf sf-provisioned-attribute-mask {
                       type attribute-mask-type;
                       default bonded;
                       description "This attribute represents the Provisioned Attribute Mask
encoding for the bonding group.";
                    }
                    leaf dsid-resequencing-warning-threshold {
                        type uint8 {
                            range "0..179 | 255";
                        }
                        units "hundred microseconds";
                        default 255;
                        description
                            "This attribute provides the DSID Resequencing Warning Threshold in
hundredMicroseconds that is to be used for all DSIDs associated with this Downstream Bonding Group.
The value of 255 indicates that the DSID Resequencing Warning Threshold is determined by the CMTS.
The value of 0 indicates that the threshold warnings are disabled.
                             When the value of DsidReseqWaitTime is not equal to 0 or 255, the CCAP
will ensure that the value of this object is either 255 or less than the value of
DsidReseqWaitTime.";
                    }
                    leaf dsid-resequencing-wait-time {
                        type uint8 {
                            range "1..180 | 255";
                        }
                        units "hundred microseconds";
                        default 255;
                        description "This attribute provides the DSID Resequencing Wait Time in
hundredMicroseconds that is to be used for all DSIDs associated with this Downstream Bonding Group.
The value of 255 indicates that the DSID Resequencing Wait Time is determined by the CMTS.";
                    }
                    list docsis-down-channel-ref {
                        key "slot ds-rf-port down-channel";
                        description "A reference to a DOCSIS downstream channel.";
                        ccap:inlineType;
                        uses docsis-down-channel-reference;
                        container yang-ext {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                    leaf downstream-bonding-mac-domain-name {
                        type leafref {
```

```
                        path "../../mac-domain/mac-domain-name";
                    }
                    description "The name of the MacDomain.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list deny-cm {
                key device-mac-address;
                leaf device-mac-address {
                    type yang:mac-address;
                    description "The MAC address of the CM that will be added to the deny
list.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list mac-domain {
                key mac-domain-name;
                min-elements 0;
                description "Mac-domain contains MAC domain level control and configuration
attributes.";
                reference
                    "DOCSIS 3.0 Operations Support System Interface Specification
                     CM-SP-OSSIv3.0-I15-110623 MdCfg Object section.";
                uses mac-domain-configuration-group;
                leaf mdd-interval {
                    type uint16 {
                        range "1..2000";
                    }
                    units milliseconds;
                    default 2000;
                    description
                        "This attribute configures the interval for the insertion of MDD
messages
                         in each downstream channel of a MAC Domain.";
                    reference
                        "DOCSIS 3.0 Operations Support System Interface Specification
                         CM-SP-OSSIv3.0-I15-110623 MdCfg Object section.";
                }
                leaf cm-status-event-control-enabled {
                    type boolean;
                    default true;
                    description "If set to 'true', this attribute enables the signaling of the
CM-Status Event reporting mechanism.";
                    reference
                        "DOCSIS 3.0 Operations Support System Interface Specification
                         CM-SP-OSSIv3.0-I15-110623 MdCfg Object section.";
                }
                leaf upstream-frequency-range {
                    type upstream-frequency-range-type;
                    default standard;
                    description "This attribute indicates in MDD messages the upstream frequency
upper band edge of an upstream Channel.";
                }
                leaf multicast-dsid-forward-enabled {
                    type boolean;
                    default true;
                    description "If set to 'true', this attribute enables the CMTS to use IP
Multicast DSID Forwarding (MDF) for the MAC domain.";
                    reference
                        "DOCSIS 3.0 Operations Support System Interface Specification
                         CM-SP-OSSIv3.0-I15-110623 MdCfg Object section.";
                }
                leaf multiple-receive-channel-mode-enabled {
                    type boolean;
                    default true;
                    description "If set to 'true', this attribute enables Downstream Channel
Bonding for the MAC Domain.";
                    reference
                        "DOCSIS 3.0 Operations Support System Interface Specification
                         CM-SP-OSSIv3.0-I15-110623 MdCfg Object section.";
                }
                leaf multiple-transmit-channel-mode-enabled {
```

```
                        type boolean;
                        default true;
                        description "If set to 'true', this attribute enables Multiple Transmit
Channel (MTC) Mode for the MAC Domain.";
                        reference
                            "DOCSIS 3.0 Operations Support System Interface Specification
                             CM-SP-OSSIv3.0-I15-110623 MdCfg Object section.";
                    }
                    leaf early-auth-encrypt-control {
                        type early-auth-encrypt-control-type;
                        default enable-eae-ranging-based-enforcement;
                        description "This attribute enables or disables early authentication and
encryption (EAE) signaling for the MAC Domain. It also defines the type of EAE enforcement in the
case that EAE is enabled.";
                    }
                    leaf tftp-proxy-enabled {
                        type boolean;
                        default true;
                        description "If set to 'true', this attribute enables TFTP Proxy
functionality for the MAC Domain.";
                        reference
                            "DOCSIS 3.0 Operations Support System Interface Specification
                             CM-SP-OSSIv3.0-I15-110623 MdCfg Object section.";
                    }
                    leaf source-address-verification-enabled {
                        type boolean;
                        default true;
                        description "If set to 'true', this attribute enables Source Address
Verification (SAV) functionality for the MAC Domain.";
                        reference
                            "DOCSIS 3.0 Operations Support System Interface Specification
                             CM-SP-OSSIv3.0-I15-110623 MdCfg Object section.";
                    }
                    leaf cm-udc-enabled {
                        type boolean;
                        default false;
                        description "If set to 'true', this attribute instructs the CMTS MAC Domain
to enable Upstream Drop Classifiers (UDC) for the CMs attempting registration in this MAC Domain.";
                        reference
                            "DOCSIS 3.0 Operations Support System Interface Specification
                             CM-SP-OSSIv3.0-I15-110623 MdCfg Object section.";
                    }
                    leaf send-udc-rules-enabled {
                        type boolean;
                        default false;
                        description
                            "If set to 'true' and when the CM signals to the CMTS 'Upstream Drop
Classifier Group ID' encodings, this attribute instructs the CMTS MAC Domain to send the Subscriber
Management Filters rules associated with the 'Upstream Drop Classifier Group ID' encodings to the CM
in the form of UDCs when the following conditions occurs:
                             - The attribute CmUdcEnabled value for this MAC Domain is set to
'true', and
                             - The CM has the UDC capability advertised as supported.
                             If there is no a single Subscriber Management Filter configured in the
CMTS for the CM's signaled UDC Group ID, the CMTS does not send UDC encodings to the CM.
                             It is vendor specific whether the CMTS maintains enforcement of the CM
signaled or default Subscriber Management Filter groups in the upstream direction.";
                        reference
                            "DOCSIS 3.0 Operations Support System Interface Specification
                             CM-SP-OSSIv3.0-I15-110623 MdCfg Object section.";
                    }
                    leaf service-type-id-list {
                        type snmp-tag-list-type;
                        description
                            "This attribute indicates the list of Service Type IDs associated with
the MAC Domain.
                             During the CM registration process the CMTS will attempt to redirect
the CM to a MAC Domain where the CM' Service Type TLV is contained in this attribute.";
                    }
                    leaf bpi2-enforce-control {
                        type bpi2-enforce-control-type;
                        default qosCfgFileWithBpi2Enabled;
                        description "This attribute indicates the level of BPI+ enforcement policies
with the MAC Domain.
";
                        reference
                            "DOCSIS 3.0 Operations Support System Interface Specification
                             CM-SP-OSSIv3.0-I15-110623 MdCfg Object section.";
```

```
                    }
                    leaf energy-mgt-1x1-enabled {
                        type boolean;
                        default false;
                        description "This attribute indicates whether the CMTS is configured for 1x1
Energy Management
                                     Mode of operation on a per MAC Domain basis. If this attribute
is set to true,
                                     the CMTS is configured for 1x1 Energy Management Mode of
operation on this MAC Domain.
                                     If this attribute is set to false, the Energy Management 1x1
Mode of operation
                                     is disabled in the CMTS on this MAC Domain.";
                        reference "DOCSIS 3.0 MAC and Upper Layer Protocols Interface Specification
                                   CM-SP-MULPIv3.0-I20-121113, Energy Management Capabilities
section.";
                    }
                    container md-bpi-config {
                        description "This object is based on DocsBpiCmtsBaseEntry table defined in
[RFC 3083].
                                     This optional object provides the configuration of the Baseline
Privacy key lifetimes for the MAC domain. If not used, the CCAP uses the defaults defined in
SysBpiCfg.";
                        reference "RFC 3083";
                        leaf default-authentication-lifetime {
                            type uint32;
                            units seconds;
                            description "The value of this object is the default lifetime, in
seconds, the CCAP assigns to a new authorization key.";
                        }
                        leaf default-tek-lifetime {
                            type uint32;
                            units seconds;
                            description "The value of this object is the default lifetime, in
seconds, the CCAP assigns to a new Traffic Encryption Key (TEK).";
                        }
                        container yang-ext {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                    list upstream-bonding-group {
                        key bonding-group-name;
                        min-elements 0;
                        description "The upstream-bonding-group object allows for the static
creation of upstream bonding groups To configure an upstream bonding group, an instance of the
UsBondingGrpCfg object is created.";
                        leaf bonding-group-name {
                            type string;
                            description "The name of the bonding group.";
                        }
                        leaf sf-provisioned-attribute-mask {
                            type attribute-mask-type;
                            default bonded;
                            description "This attribute represents the Provisioned Attribute Mask
encoding for the bonding group.";
                        }
                        list upstream-logical-channel-ref {
                            key "slot us-rf-port upstream-physical-channel upstream-logical-
channel";
                            min-elements 0;
                            description "This element configures an upstream logical channel to be
part of this upstream bonding group.";
                            uses upstream-logical-channel-reference;
                            container yang-ext {
                                ccap:extensionPoint; //different pyang flags impact use of this hint
                                description "node for vendor YANG extensions";
                            }
                        }
                        container yang-ext {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                    list rcc-configuration {
                        key "rcp-id rcc-cfg-id";
                        min-elements 0;
                        description
```

```
                            "This object is based on the RccCfg object defined in [OSSIv3.0] and is
used with the following modification: The MdIfIndex attribute has been removed and replaced by the
named association between MdCfg and RccCfg.
                             This object creates static Receive Channel Configurations for specific
downstream channel configurations, identifies the scope of the Receive Channel Configuration (RCC),
and provides a top level container for the Receive Module and Receive Channel objects. The CCAP
selects an instance of this object to assign to a CM when it registers.
                             This object supports the creation and deletion of multiple instances.";
                        reference
                            "DOCSIS 3.0 Operations Support System Interface Specification
                             CM-SP-OSSIv3.0-I15-110623 RccCfg Object section.";
                        leaf rcp-id {
                            type ccap-octet-data-type {
                                length "10";
                            }
                            description "This key represents the 'Receive Channel Profile
Identifier' (RCP-ID) configured for the MAC Domain indicated by this instance.";
                            reference
                                "DOCSIS 3.0 Operations Support System Interface Specification
                                 CM-SP-OSSIv3.0-I15-110623 RccCfg Object section.";
                        }
                        leaf rcc-cfg-id {
                            type uint32 {
                                range "1..4294967295";
                            }
                            description "This key denotes an RCC combination assignment for a
particular RcpId and is unique per combination of MAC Domain and RcpId.";
                            reference
                                "DOCSIS 3.0 Operations Support System Interface Specification
                                 CM-SP-OSSIv3.0-I15-110623 RccCfg Object section.";
                        }
                        leaf vendor-specific {
                            type ccap-octet-data-type {
                                length "0..504";
                            }
                            description "This attribute contains vendor-specific information of the
CM Receive Channel configuration.";
                            reference
                                "DOCSIS 3.0 Operations Support System Interface Specification
                                 CM-SP-OSSIv3.0-I15-110623 RccCfg Object section.";
                        }
                        leaf description {
                            type string {
                                length "0..15";
                            }
                            default " ";
                            description "This attribute contains a human-readable description of the
CM RCP Configuration.";
                        }
                        list receive-channel-configuration {
                            key receive-channel-id;
                            min-elements 0;
                            description
                                "The Receive Channel Configuration object permits an operator to
configure how CMs registered with certain Receive Channel Profiles will configure the Receive
Channels within their profile.
                                 When a CM registers with a Receive Channel Profile (RCP) for which
all Receive Channel Indices (RcIds) are configured in the Receive Module object and all Receive
Channels are configured within this object, the CCAP should use the configuration within these
objects to set the Receive Channel Configuration returned to the CM in a REG-RSP message.
                                 The CCAP may require configuration of all pertinent Receive Module
and Receive Channel instances in order to register a CM that reports a Receive Channel Profile
(RCP), including any standard Receive Channel Profiles.
                                 If the CM reports multiple RCPs, and Receive Module and Receive
Channel objects have instances for more than one RCP, the particular RCP selected by the CCAP is not
specified. A CCAP is not restricted to assigning Receive Modules based only on the contents of this
object.";
                            leaf receive-channel-id {
                                type uint8 {
                                    range "1..255";
                                }
                                description "This key represents an identifier for the parameters of
the Receive Channel instance within the Receive Channel Profile.";
                                reference
                                    "DOCSIS 3.0 Operations Support System Interface Specification
                                     CM-SP-OSSIv3.0-I15-110623 RxChCfg Object section.";
                            }
                            leaf primary-downstream-indicator {
```

```
                                type boolean;
                                default false;
                                description "If set to 'true', this attribute indicates the Receive
Channel is to be the primary-capable downstream channel for the CM receiving this RCC. Otherwise,
the downstream channel is to be a non-primary-capable channel.";
                                reference
                                    "DOCSIS 3.0 Operations Support System Interface Specification
                                     CM-SP-OSSIv3.0-I15-110623 RxChCfg Object section.";
                            }
                            leaf rc-rm-connectivity-identifier {
                                type leafref {
                                    path "../../receive-module-configuration/receive-module-id";
                                }
                                description "This attribute indicates the Receive Module (via the
RmId from the RxModule object) to which this Receive Channel connects. If this object contains a
zero value (and thus no Receive Channel Connectivity), the Receive Channel Connectivity TLV is
omitted from the RCC.";
                                reference
                                    "DOCSIS 3.0 Operations Support System Interface Specification
                                     CM-SP-OSSIv3.0-I15-110623 RxChCfg Object section.";
                            }
                            container docsis-down-channel-ref {
                                description "A reference to a DOCSIS downstream channel.";
                                ccap:inlineType;
                                uses docsis-down-channel-reference;
                                container yang-ext {
                                    ccap:extensionPoint; //different pyang flags impact use of this
hint
                                    description "node for vendor YANG extensions";
                                }
                            }
                            container yang-ext {
                                ccap:extensionPoint; //different pyang flags impact use of this hint
                                description "node for vendor YANG extensions";
                            }
                        }
                        list receive-module-configuration {
                            key receive-module-id;
                            min-elements 0;
                            description
                                "The Receive Module Configuration object permits an operator to
configure how CMs with certain RCPs will configure the Receive Modules within their profile upon CM
registration.
                                 When a CM registers with an RCP for which all Receive Module
Indices (RmIds) are configured in this object and all Receive Channels are configured within the
Receive Channel (RxCh) object, the CCAP should use the configuration within these objects to set the
Receive Channel Configuration assigned to the CM in a REG-RSP message.
                                 The CCAP may require configuration of all pertinent Receive Module
and Receive Channel instances in order to register a CM that reports a Receive Channel Profile.
                                 If the CM reports multiple RCPs, and Receive Module and Receive
Channel objects have instances for more than one RCP reported by the CM, the particular RCP selected
by the CCAP is not specified. A CCAP is not restricted to assigning Receive Modules based only on
the contents of this object.";
                            leaf receive-module-id {
                                type uint8 {
                                    range "1..255";
                                }
                                description "This key represents an identifier of a Receive Module
instance within the Receive Channel Profile.";
                                reference
                                    "DOCSIS 3.0 Operations Support System Interface Specification
                                     CM-SP-OSSIv3.0-I15-110623 RxModuleCfg Object section.";
                            }
                            leaf rm-rm-connectivity-id {
                                type leafref {
                                    path "../../receive-module-configuration/receive-module-id";
                                }
                                description
                                    "This attribute represents the higher level (i.e. closer to RF)
Receive Module to which this Receive Module connects. If this object contains a zero value (and thus
no Receive Module Connectivity), the Receive Module Connectivity TLV is omitted from the RCC.
                                     Within a single instance of the RxModule object, the
RmRmConnectivityId attribute cannot contain the same value as the RmId attribute. The
RmRmConnectivityId attribute points to a separate RxModule object instance with the same value of
RccCfgId.";
                                reference
                                    "DOCSIS 3.0 Operations Support System Interface Specification
                                     CM-SP-OSSIv3.0-I15-110623 RxModuleCfg Object section.";
```

```
                            }
                            leaf first-center-frequency {
                                type uint32;
                                units hertz;
                                default 0;
                                description "This attribute represents the center frequency, in Hz,
and a multiple of 62500, that indicates the low frequency channel of the Receive Module, or 0 if not
applicable to the Receive Module.";
                                reference
                                    "DOCSIS 3.0 Operations Support System Interface Specification
                                     CM-SP-OSSIv3.0-I15-110623 RxModuleCfg Object section.";
                            }
                            container yang-ext {
                                ccap:extensionPoint; //different pyang flags impact use of this hint
                                description "node for vendor YANG extensions";
                            }
                        }
                        container yang-ext {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                    choice cmts-mac-interface {
                        case cmts-mac-interface-config {
                            description "This object describes the attributes of each CMTS MAC
interface.";
                            reference "RFC 4546 docsIfCmtsMacTable";
                            leaf sync-interval {
                                type uint8 {
                                    range "1..200";
                                }
                                units milliseconds;
                                mandatory true;
                                description "The interval between CMTS transmission of successive
SYNC messages at this interface.";
                                reference "RFC 4546";
                            }
                            leaf ucd-interval {
                                type uint16 {
                                    range "1..2000";
                                }
                                units milliseconds;
                                mandatory true;
                                description "The interval between CMTS transmission of successive
Upstream Channel Descriptor messages for each upstream channel at this interface.";
                                reference "RFC 4546";
                            }
                            leaf invited-ranging-attempts {
                                type uint16 {
                                    range "0..1024";
                                }
                                units attempts;
                                mandatory true;
                                description "The maximum number of attempts to make on invitations
for ranging requests.  A value of zero means the system should attempt to range forever.";
                                reference "RFC 4546";
                            }
                            leaf im-insertion-interval {
                                type uint32 {
                                    range "0..2147483647";
                                }
                                units HundredOfSeconds;
                                mandatory true;
                                description "The amount of time to elapse between each broadcast
initial maintenance grant.  Broadcast initial maintenance grants are used to allow new cable modems
to join the network.  Zero indicates that a vendor-specific algorithm  is used instead of a fixed
time.  The maximum amount of time permitted by the specification is 2 seconds.";
                                reference "RFC 4546";
                            }
                            leaf docsis11-concatenation-enabled {
                                type boolean;
                                default true;
                                description "Enables and disables DOCSIS 1.1 concatenation.";
                            }
                            leaf docsis11-fragmentation-enabled {
                                type boolean;
                                default true;
                                description "Enables and disables DOCSIS 1.1 fragmentation.";
```

```
                        }
                        container yang-ext5 {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                }
                list upstream-physical-channel-ref {
                    key "slot us-rf-port upstream-physical-channel";
                    min-elements 0;
                    description "A reference to an upstream physical channel.";
                    uses upstream-physical-channel-reference;
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list non-primary-capable-ds {
                    key "slot ds-rf-port down-channel";
                    min-elements 0;
                    description "A reference to a non-primary capable DOCSIS downstream channel.
                                 Some CCAP implementations may implement the association of non
primary capable downstream channels with MAC domain indirectly, based on RF plant topology
configuration. In such a case CCAP device may ignore configuration settings communicated through the
label Non-PrimaryCapableDs";

                    uses docsis-down-channel-reference;
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list primary-capable-ds {
                    key "slot ds-rf-port down-channel";
                    min-elements 0;
                    description "A reference to a primary capable DOCSIS downstream channel.";
                    uses docsis-down-channel-reference;
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            container yang-ext {
                ccap:extensionPoint; //different pyang flags impact use of this hint
                description "node for vendor YANG extensions";
            }
        }
        container docs-multicast-authorization {
            description "The docs-multicast-authorization group of configuration elements allows
for the configuration of DOCSIS Multicast. They are based on the configuration elements from
[OSSIv3.0]. ";
            container control {
                description "This object defines the CCAP global behavior for Multicast
Authorization. Some parameters are included as part of the CM configuration process. In absence of
those parameters, default values defined by attributes of this object are used.

";
                reference
                    "DOCSIS 3.0 Operations Support System Interface Specification
                     CM-SP-OSSIv3.0-I15-110623 Ctrl Object section.";
                leaf enable {
                    type enable-type;
                    default disable;
                    description "This attribute enables the enforcement of Multicast
Authorization feature.";
                }
                leaf default-profile-name-list {
                    type snmp-tag-list-type;
                    default " ";
                    description "This attribute indicates one or more Multicast Authorization
Profiles that are used by the CMTS when CMs register with no Multicast Join Authorization encodings
in the REG-REQ-(MP). When IP Multicast Authorization is enforced, this attribute provides the
default set of Multicast Authorization Profiles the CMTS enforces for a CM in case the CM did not
```

```
signal a set of profiles during the registration process. If the Default Multicast Authorization
Group Name is a -zero-length string, the DefAction attribute determines whether a join request is
authorized. If the CMTS supports more than one profile name as a default, the CMTS enforces each of
the profiles in order of occurrence until the maximum number of profiles is reached.";
                    }
                    leaf default-action {
                        type authorization-action-type;
                        default deny;
                        description "This attribute defines the default authorization action when no
IP Multicast Session Rule is determined to match a client's IP multicast JOIN request.";
                    }
                    leaf default-max-number-sessions {
                        type uint16;
                        default 0;
                        description "This attribute indicates the default maximum number of
multicast sessions that clients reached through a particular CM are allowed to join. A DefMaxNumSess
value of 0 indicates that no dynamic joins are permitted. A Maximum Multicast Sessions Encoding
value of 65535 (the largest valid value) indicates that the CMTS permits any number of sessions to
be joined by clients reached through the CM.
";
                        reference
                            "DOCSIS 3.0 MAC and Upper Layer Protocols Interface Specification
                             CM-SP-MULPIv3.0-I16-110623 Maximum Multicast Sessions section.";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list profiles {
                    key mcast-auth-profile-name;
                    min-elements 0;
                    description
                        "This object contains the description of the Multicast Authorization
profiles for administrative purposes.
                         This object supports the creation and deletion of multiple instances.
                         Creation of a new instance of this object requires the Description
attribute to be set.";
                    reference
                        "DOCSIS 3.0 Operations Support System Interface Specification
                         CM-SP-OSSIv3.0-I15-110623 Profiles Object section.";
                    leaf mcast-auth-profile-name {
                        type string {
                            length "1..15";
                        }
                        description "This attribute is a unique name or identifier for a Multicast
Authorization Profile.";
                    }
                    leaf description {
                        type string;
                        mandatory true;
                        description "This attribute is a human readable description of the Multicast
Authorization Profile.";
                    }
                    list session-rule {
                        key "id session-rule-name";
                        min-elements 1;
                        description
                            "This object defines Operator configured profiles to be matched during
the authorization process.
                             This object supports the creation and deletion of multiple instances.
                             Creation of a new instance of this object requires the following
attributes to be set:
                             - SrcPrefixAddr
                             - GrpPrefixAddr";
                        reference
                            "DOCSIS 3.0 Operations Support System Interface Specification
                             CM-SP-OSSIv3.0-I15-110623 ProfileSessRule Object section.";
                        leaf session-rule-name {
                            type string {
                                length "1..15";
                            }
                            description "This attribute is a unique name that associates the IP
Multicast Authorization Profile Name Subtype encoding signaled by CMs with the a set of Multicast
Authorization Profile Session Rules.";
                        }
                        leaf id {
                            type uint32;
```

```
                            description "This attribute provides a unique identifier for each CMTS
configured Multicast Authorization Profile Session rule within a Multicast Authorization Profile
Name.";
                        }
                        leaf priority {
                            type uint32;
                            default 0;
                            description "This attribute configures the rule priority for the static
session rule. Higher values indicate a higher priority. If more than one session rule matches a
joined session, the session rule with the highest rule priority determines the authorization
action.";
                        }
                        uses ssm-group;
                        leaf authorization-action {
                            type authorization-action-type;
                            default deny;
                            description
                                "This attribute specifies the authorization action for a session
join attempt that matches the session rule.";
                        }
                        container yang-ext {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            choice docsis-if {
                case docs-if {
                    description "The docs-if object is the container for DOCSIS interface
configuration objects.";
                    list modulation-profile {
                        key modulation-index;
                        min-elements 1;
                        description "This object is based on the docsIfCmtsModulationTable defined
in [RFC 4546].";
                        reference "RFC 4546";
                        leaf modulation-index {
                            type uint32 {
                                range "1..2147483647";
                            }
                            description "An index into the Channel Modulation table representing a
group of Interval Usage Codes, all associated with the same channel.";
                        }
                        list interval-usage-code {
                            key usage-code;
                            min-elements 1;
                            max-elements 14;
                            description "This object allows a list of interval usage codes to be
associated with a single modulation profile. It is based on the docsIfCmtsModulationTable defined in
[RFC 4546].
                                 Reference: [RFC 4546] docsIfCmtsModulationTable.";
                            leaf usage-code {
                                type modulation-interval-usage-code-type;
                                mandatory true;
                                description "An index into the Channel Modulation table that, when
grouped with other Interval Usage Codes, fully instantiates all modulation sets for a given upstream
channel.";
                            }
                            leaf modulation {
                                type modulation-type;
                                default qpsk;
                                description "The modulation type used on this channel.";
                            }
                            leaf preamble-length {
                                type preamble-length-type;
                                mandatory true;
                                description "The preamble length for this modulation profile in
bits.";
                            }
```

```
                            leaf differential-encoding {
                                type boolean;
                                default false;
                                description "Specifies whether or not differential encoding is used
on this channel.";
                            }
                            leaf fec-error-correction {
                                type fec-error-correction-type;
                                default 0;
                                description "The number of correctable errored bytes (t) used in
forward error correction code.";
                            }
                            leaf fec-codeword-length {
                                type fec-codeword-length-type;
                                default 32;
                                description
                                    "The number of data bytes (k) in the forward error correction
codeword.
                                     This object is not used if fec-error-correction is zero.";
                            }
                            leaf scrambler-seed {
                                type scrambler-seed-type;
                                default 0;
                                description "The 15-bit seed value for the scrambler polynomial.";
                            }
                            leaf max-burst-size {
                                type uint8;
                                units mini-slots;
                                mandatory true;
                                description "The maximum number of mini-slots that can be
transmitted during this channel's burst time. Default value is 0, except for shortData, where it is
8.";
                            }
                            leaf last-codeword-shortened {
                                type boolean;
                                default true;
                                description "Indicates whether the last FEC codeword is truncated.";
                            }
                            leaf scrambler {
                                type boolean;
                                default false;
                                description "Indicates whether the scrambler is employed.";
                            }
                            leaf byte-interleaver-depth {
                                type uint32;
                                default 1;
                                description "ATDMA Byte Interleaver Depth (Ir).";
                            }
                            leaf byte-interleaver-block-size {
                                type uint32;
                                default 18;
                                description "ATDMA Byte Interleaver Block size (Br).";
                            }
                            leaf preamble {
                                type preamble-type;
                                default qpsk0;
                                description "Preamble type for DOCSIS 2.0 bursts.  ";
                            }
                            leaf tcm-error-correction-on {
                                type boolean;
                                default false;
                                description "Trellis Code Modulation (TCM) On/Off.";
                            }
                            leaf scdma-interleaver-step-size {
                                type scdma-interleaver-step-size-type;
                                default 1;
                                description "S-CDMA Interleaver step size.";
                            }
                            leaf scdma-spreader-enable {
                                type boolean;
                                description "S-CDMA spreader.  Default value for IUC 3 and 4 is OFF;
for all other IUCs it is ON.";
                            }
                            leaf scdma-subframe-codes {
                                type scdma-subframe-codes-type;
                                default 1;
                                description "S-CDMA sub-frame size.";
                            }
```

```
                            leaf channel-type {
                                type us-channel-type;
                                default tdma;
                                description "Describes the modulation channel type for this
modulation entry.";
                            }
                            container yang-ext {
                                ccap:extensionPoint; //different pyang flags impact use of this hint
                                description "node for vendor YANG extensions";
                            }
                        }
                        container yang-ext {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                    container yang-ext6 {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
            }
            container docs-packet-cable {
                description "The docs-packet-cable object is the container for PacketCable
configuration objects.";
                choice packet-cable {
                    case packet-cable-config {
                        description "This object is used for configuring PacketCable and PCMM
services on the CCAP.";
                        leaf packet-cable-enabled {
                            type boolean;
                            default false;
                            description "This configuration attribute allows the operator to enable
PacketCable services on the CCAP.";
                        }
                        leaf pcmm-enabled {
                            type boolean;
                            default false;
                            description "This configuration attribute allows the operator to enable
PacketCable Multimedia services on the CCAP.";
                        }
                        leaf pc-t0-timer {
                            type uint8;
                            units seconds;
                            default 30;
                            description "This configuration attribute allows the operator to define
the value in seconds for the PacketCable T0 timer.";
                        }
                        leaf pc-t1-timer {
                            type uint32;
                            units seconds;
                            default 200;
                            description "This configuration attribute allows the operator to define
the value in seconds for the PacketCable T1 timer.";
                        }
                        leaf pc-t7-timer {
                            type uint32;
                            units seconds;
                            default 200;
                            description
                                "This attribute allows for the setting of the Timeout for Admitted
QoS Parameters for the service flow to the value specified for this timer. In the case of a flow
with multiple sub-flows, the flow's Timeout for Admitted QoS
                                 Parameters is set to the value of timer T7 from the most recently
received Gate-Set message for any subflow on the flow. The Timeout for Admitted QoS Parameters
limits the period of time that the CMTS holds resources for a service flow's Admitted QoS Parameter
Set while they are in excess of its Active QoS Parameter Set. ";
                            reference "PacketCable 1.5 Dynamic Quality-of-Service Specification,
PKT-SP-DQOS1.5-I04-090624";
                        }
                        leaf pc-t8-timer {
                            type uint8;
                            units seconds;
                            default 0;
                            description "This attribute configures the Timeout for Active QoS
Parameters for the service flow to the value specified in for this timer. In the case of a flow with
multiple sub-flows, the flow's Timeout for Active QoS Parameters is set to the value of timer T8
```

```
from the most recently received Gate-Set message for any sub-flow on the flow. The Timeout for
Active QoS Parameters limits the period of time resources remain unused on an active service flow.
    ";
                            reference "PacketCable 1.5 Dynamic Quality-of-Service Specification,
PKT-SP-DQOS1.5-I04-090624";
                        }
                        leaf pcmm-t1-timer {
                            type uint8;
                            units seconds;
                            default 200;
                            description "This configuration attribute allows the operator to define
the value in seconds for the PacketCable Multimedia T1 timer.";
                        }
                        leaf cmts-gate-id-value {
                            type uint32 {
                                range "0..16383";
                            }
                            mandatory true;
                            description "This configuration attribute allows the operator to define
the value for the CMTS Id portion of PCMM GateIds.";
                        }
                        leaf tos {
                            type int8 {
                                range "-1 | 0..63";
                            }
                            mandatory true;
                            description "This configuration attribute allows the operator to define
the value for the Tos bits in outgoing COPS messages.";
                        }
                        leaf cops-connection-threshold {
                            type uint32;
                            units "connections/15 mins";
                            mandatory true;
                            description "This configuration attribute allows the operator to define
the threshold number of COPS connections per 15 minute interval.";
                        }
                        leaf control-point-discovery-enabled {
                            type boolean;
                            default false;
                            description "This attribute enables or disables the Control point
Discovery functionality described in the PacketCable Specifications.";
                        }
                        container yang-ext7 {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                }
                choice pc-event {
                    case pc-event-config {
                        description "This object configures event messaging for PacketCable.";
                        leaf retry-timer {
                            type uint16 {
                                range "10..10000";
                            }
                            units milliseconds;
                            default 3000;
                            description "This configuration attribute allows the configuration of
the number of milli-seconds the CCAP should wait before sending a message that was not
acknowledged.";
                        }
                        leaf retry-limit {
                            type uint8 {
                                range "0..9";
                            }
                            default 3;
                            description "This configuration attribute allows the configuration of
the number of times the CCAP should retry when sending a message.";
                        }
                        leaf batch-size {
                            type uint32;
                            mandatory true;
                            description "This configuration attribute allows the configuration of
the number of records the CCAP should bundle in a single message to a billing or Record Keeping
Server (RKS).";
                        }
                        leaf max-age {
                            type uint32;
```

```
                        units seconds;
                        mandatory true;
                        description "This object defines the max age of messages to be sent to a
RKS or billing server.";
                    }
                    leaf billing-events {
                        type boolean;
                        default false;
                        description "This attribute tells the CCAP if it needs to send billing
events to a billing server/RKS.";
                    }
                    container yang-ext8 {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }

            }
        }
        container docs-dsg {
            description "The docs-dsg object is the container for DSG configuration objects.";
            list dsg-timer-config {
                key timer-config-index;
                min-elements 0;
                description
                    "This configuration object is based on the dsgIfTimerTable defined in [DSG]
and will be used with modifications for CCAP.
                     The DSG Timer Table contains timers that are sent to the DSG client(s) via
the DCD message. ";
                reference
                    "DOCSIS Set-top Gateway (DSG) Interface Specification CM-SP-DSG-I18-110623
                     DOCSIS Set-top Gateway Agent MIB Definition section";
                leaf timer-config-index {
                    type uint32 {
                        range "1..4294967295";
                    }
                    description "The index for timer-config list.";
                }
                leaf init-t-dsg-1 {
                    type uint16 {
                        range "1..65535";
                    }
                    units seconds;
                    default 2;
                    description "Initialization Timeout. This is the timeout period in seconds
for the DSG packets during initialization of the DSG client.";
                }
                leaf oper-t-dsg-2 {
                    type uint16 {
                        range "1..65535";
                    }
                    units seconds;
                    default 600;
                    description "Operational Timeout. This is the timeout period in seconds for
the DSG packets during normal operation of the DSG client.";
                }
                leaf two-way-t-dsg-3 {
                    type uint16;
                    units seconds;
                    default 300;
                    description "Two-way retry timer. This is the retry timer that determines
when the DSG client attempts to reconnect with the DSG Agent and established two-way connectivity.
Default value is 300 seconds. The value 0 indicates that the client will continuously retry two-way
operation";
                }
                leaf one-way-t-dsg-4 {
                    type uint16;
                    units seconds;
                    default 1800;
                    description "One-way retry timer. The retry timer that determines when the
client attempts to rescan for a DOCSIS downstream channel that contains DSG packets after a
TimerTdsg1 or TimerTdsg2 timeout. Default value is 1800 seconds. The value 0 indicates that the
client will immediately begin scanning upon TimerTdsg1 or TimerTdsg2 timeout.";
                }
```

```
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list dsg-downstream {
                    key dsg-downstream-index;
                    min-elements 0;
                    description "The dsg-downstream object represents an individual downstream
channel for DSG configuration purposes. This configuration object is based on the
dsgIfDownstreamTable defined in [DSG] and will be used with modifications for CCAP.";
                    leaf dsg-downstream-index {
                        type uint32;
                        description "This is the key for an instance of this object. In [DSG], this
was the channel ifIndex. Here, the channel is identified uniquely by the docsis-down-channel-ref.";
                    }
                    leaf enable-dcd {
                        type boolean;
                        mandatory true;
                        description "This attribute is used to enable or disable DCD messages to be
sent on this downstream channel. The value is always true for those downstreams that contain DSG
tunnels.";
                    }
                    container docsis-down-channel-ref {
                        description "This is a reference to a DOCSIS down channel instance.";
                        ccap:inlineType;
                        uses docsis-down-channel-reference;
                        container yang-ext {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                    leaf timer-config-index {
                        type leafref {
                            path "../../dsg-timer-config/timer-config-index";
                        }
                        description "Reference to the timer-config index. Zero means not
configured.";
                    }
                    leaf vendor-param-id {
                        type leafref {
                            path "../../vendor-parameters-list/vendor-param-id";
                        }
                        description "Reference to the vendor-parameters id. Zero means not
configured.";
                    }
                    leaf dsg-channel-list-index {
                        type leafref {
                            path "../../dsg-channel-list/dsg-channel-list-index";
                        }
                        description "Reference to dsg-channel-list enumerating the frequencies
carrying DSG tunnels. Zero means not configured.";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list dsg-channel-list {
                    key "dsg-channel-list-index";
                    min-elements 0;
                    description
                        "This configuration object is based on the dsgIfChannelListTable defined in
[DSG] and will be used with modifications for CCAP.
                         The DownChanList object allows for configuration of a list of one or
multiple downstream frequencies that are carrying DSG tunnel(s). This configuration object has been
modified from the DSG Specification definitions.";
                    reference
                        "DOCSIS Set-top Gateway (DSG) Interface Specification CM-SP-DSG-I18-110623
                         DOCSIS Set-top Gateway Agent MIB Definition section";
                    leaf dsg-channel-list-index {
                        type uint32 {
                            range "1..4294967295";
                        }
                        description "The index of the dsg channel list.";
                    }
                    list dsg-channel {
                        key "dsg-channel-index";
```

```
                    min-elements 1;
                    description "The frequencies carrying DSG tunnels for some RF connector.";
                    leaf dsg-channel-index {
                        type uint32;
                        description "The local index of the channel.";
                    }
                    leaf channel-downstream-frequency {
                        type uint32 {
                            range "0..1000000000";
                        }
                        units Hz;
                        default 0;
                        description "The channel-downstream-frequency attribute represent a
frequency of a downstream channel carrying DSG information. Frequency is a multiple of 62500 Hz, per
[DSG]";
                        reference
                            "DOCSIS Set-top Gateway (DSG) Interface Specification CM-SP-DSG-I18-
110623
                             DOCSIS Set-top Gateway Agent MIB Definition section";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list tunnel-group-to-channel-list {
                key "tunnel-group-index";
                min-elements 0;
                description
                    "This configuration object is based on the dsgIfTunnelGrpToChannelTable
defined in [DSG] and will be used with modifications for CCAP.
                     The TunnelGrpToChannel object permits association of a group of tunnels to
one or more downstream channels. This configuration object has been modified from the DSG
Specification definitions.";
                reference
                    "DOCSIS Set-top Gateway (DSG) Interface Specification CM-SP-DSG-I18-110623
                     DOCSIS Set-top Gateway Agent MIB Definition section";
                leaf tunnel-group-index {
                    type uint32;
                    description "This attribute is the key for this tunnel group.";
                }
                list tunnel-group-channel {
                    key "tunnel-group-channel-index";
                    min-elements 1;
                    description "The channels in the tunnel group and the tunnel group
parameters for each channel.";
                    leaf tunnel-group-channel-index {
                        type uint32;
                        description "This attribute configures the linkage to a specific
DownChan instance that will carry the DSG tunnels in this tunnel group.";
                    }
                    leaf rule-priority {
                        type uint8 {
                            range "0..255";
                        }
                        default 0;
                        description "The DSG rule priority determines the order in which the DSG
rules are applied by the DSG client. The default value is 0, which is the lowest priority.";
                    }
                    leaf vendor-param-id {
                        type leafref {
                            path "../../../vendor-parameters-list/vendor-param-id";
                        }
                       description "Reference to the Vendor Parameters Id. Zero means not
configured.";
                    }
                    leaf dsg-downstream-index {
                        type leafref {
                            path "../../../dsg-downstream/dsg-downstream-index";
                        }
                        mandatory true;
                        description "Reference to the DSG Down Channel Index.";
                    }
```

```
                        container yang-ext {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list dsg-tunnel-config {
                    key dsg-tunnel-config-index;
                    min-elements 0;
                    description
                        "A dsg-tunnel-config object allows the operator to configure DSG tunnels.
Each DSG Tunnel represents a stream of packets delivered to a DSG Client in a set-top device and is
configured with a single destination MAC address.
                         This configuration object is based on the dsgIfTunnelTable defined in [DSG]
and is used with modifications.";
                    reference
                        "DOCSIS Set-top Gateway (DSG) Interface Specification CM-SP-DSG-I18-110623
                         DOCSIS Set-top Gateway Agent MIB Definition section";
                    leaf dsg-tunnel-config-index {
                        type uint32;
                        description "This attribute is the index for a tunnel that could be
associated to one or more downstream channel that carries DSG tunnels.";
                    }
                    leaf tunnel-grp-index {
                        type leafref {
                            path "../../tunnel-group-to-channel-list/tunnel-group-index";
                        }
                        description "Reference to the DSG tunnel-grp containing this tunnel.";
                    }
                    leaf mac-address {
                        type yang:mac-address;
                        mandatory true;
                        description "This attribute configures the DSG tunnel destination MAC
address.";
                    }
                    leaf client-id-list-index {
                        type leafref {
                            path "../../client-id-config-list/client-id-list-index";
                        }
                        description "Reference to one or more objects in client-id-config. Zero
means not configured.";
                    }
                    leaf service-class-name {
                        type leafref {
                            path "/ccap/docsis/docs-qos/service-class/service-class-name";
                        }
                        mandatory true;
                        description "Reference to a service-class name.";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list dsg-classifier {
                    key dsg-classifier-id;
                    min-elements 0;
                    description "This configuration object is based on the dsgIfClassifierTable
defined in [DSG] and will be used with modifications for CCAP.";
                    reference
                        "DOCSIS Set-top Gateway (DSG) Interface Specification CM-SP-DSG-I18-110623
                         DOCSIS Set-top Gateway Agent MIB Definition section";
                    leaf dsg-classifier-id {
                        type uint16 {
                        range "1..65535";
                        }
                        description "This attribute configures the linkage between the DSG tunnel
for which this classifier will apply.";
                    }
                    leaf tunnel-index {
                        type leafref {
                            path "../../dsg-tunnel-config/dsg-tunnel-config-index";
                        }
                        description "Reference to the DSG tunnel using this classifier.";
```

```
                    }
                    leaf priority {
                        type uint8;
                        default 0;
                        description "This attribute is used to configure the DSG rule priority that
determines the order of which channel and its associated UCIDs should be applied by the DSG client.
The default value is 0, which is the lowest priority.";


                    }
                    leaf source-ip {
                        type inet:ipv4-prefix;
                        mandatory true;
                        description "This attribute configures the source prefix for the DSG tunnel.
Currently, the CCAP only supports IPv4 addresses for DSG tunnels, per [DSG]";
                        reference
                            "DOCSIS Set-top Gateway (DSG) Interface Specification CM-SP-DSG-I18-
110623
                             DOCSIS Set-top Gateway Agent MIB Definition section";
                    }
                    leaf destination-ip {
                        type inet:ipv4-address;
                        mandatory true;
                        description "This attribute configures the destination IP address for the
DSG tunnel. Currently, the CCAP only supports Ipv4 addresses for DSG tunnels, per [DSG].";
                        reference
                            "DOCSIS Set-top Gateway (DSG) Interface Specification CM-SP-DSG-I18-
110623
                             DOCSIS Set-top Gateway Agent MIB Definition section";
                    }
                    leaf destination-port-start {
                        type inet:port-number;
                        default 0;
                        description "This attribute configures the inclusive lower bound of the
transport-layer source port range that is to be matched.";
                    }
                    leaf destination-port-end {
                        type inet:port-number;
                        default 65535;
                        description "This attribute the inclusive higher bound of the transport-
layer source port range that is to be matched.";
                    }
                    leaf include-in-dcd {
                        type boolean;
                        default true;
                        description "Indicates whether or not this DSG classifier will be sent in
DCD messages for use as a Layer-3 and Layer-4 packet filter by the DSG eCM.";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list vendor-parameters-list {
                    key "vendor-param-id";
                    min-elements 0;
                    description
                        "This configuration object is based on the dsgIfVendorParamTable defined in
[DSG] and will be used without modification for CCAP.
                         The DSG Vendor Parameter Table allows vendors to send specific parameters
to the DSG clients within a DSG rule or within the DSG Configuration block in a DCD message.";
                    reference
                        "DOCSIS Set-top Gateway (DSG) Interface Specification CM-SP-DSG-I18-110623
                         DOCSIS Set-top Gateway Agent MIB Definition section";
                    leaf vendor-param-id {
                        type uint32 {
                            range "1..4294967295";
                        }
                        description "This attribute represents a key to the vendor-parameters
object. ";
                    }
                    list vendor-param {
                        key "vendor-index";
                        min-elements 1;
                        description "A single vendor parameter.";
                        leaf vendor-index {
                            type uint32;
```

```
                            description "The vendor-specific identifier configured for each instance
of the pair of VendorOui and VendorValue attributes.";
                        }
                        leaf vendor-oui {
                            type ccap-octet-data-type {
                                length "6";
                            }
                            mandatory true;
                            description "The vendor-oui attribute allows for the configuration of
vendor-assigned organization unique ID (OUI).";
                        }
                        leaf vendor-value {
                            type ccap-octet-data-type {
                                length "0..100";
                            }
                            mandatory true;
                            description "This attribute represents a vendor-specific parameter value
in the form of an octet string.";
                        }
                        container yang-ext {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list client-id-config-list {
                    key "client-id-list-index";
                    min-elements 0;
                    description
                        "This configuration object is based on the dsgIfClientIdTable defined in
[DSG] and will be used with modifications for CCAP.
                         The Client Identification object contains the client identification type
and value. It also contains the vendor-specific parameter identification. There could be multiple
client ids associated to a tunnel, grouped by the ListIndex.";
                    reference
                        "DOCSIS Set-top Gateway (DSG) Interface Specification CM-SP-DSG-I18-110623
                         DOCSIS Set-top Gateway Agent MIB Definition section";
                    leaf client-id-list-index {
                        type uint32 {
                            range "1..4294967295";
                        }
                        description "This attribute identifies a specific DSG client list.";
                    }
                    list dsg-client {
                        key "client-id-index";
                        min-elements 1;
                        description "A single DSG client.";
                        leaf client-id-index {
                            type uint32;
                            description "This attribute identifies a single client on a DSG client
list.";
                        }
                        leaf dsg-client-id-type {
                            type dsg-client-id-class-type;
                            default broadcast;
                            description "The Client Identification type.";
                        }
                        leaf client-id-value {
                            type ccap-octet-data-type {
                                length "12";
                            }
                            default 000000000000;
                            description "The Client Identification Value. The content depends on the
value of the dsgIfClientIdType. For dsgIfClientIdType of a type broadcast(1), this object will have
a 16-bit value whether or not it is a length 0 or length 2 broadcast ID. If the value is 0, then the
encoded Type Length Value Attribute (TLV) in the DCD would be the original, zero length, broadcast
ID. If the value is specified in Table 5-2 of [DSG], then the TLV in the DCD would be a length 2
broadcast ID followed by the value. For ClientIdType of a type macAddress(2), this object is a well-
known MAC address. For ClientIdType of a type caSystemId(3), this object is a CA System ID. For
ClientIdType of a type applicationId(4), this object is an application ID. Client IDs representing
types broadcast(1), caSystemId(3) or applicationId(4) are encoded in DCD messages as Unsigned
integers and configured in this object as 6 octet string with the 2 LSB for the client ID value,
e.g. an applicationId 2048 (0x0800) is encoded as '000000000800'h.";
                        }
```

```
                        leaf vendor-parameters-id {
                            type leafref {
                                path "../../../vendor-parameters-list/vendor-param-id";
                            }
                            description "Reference to Vendor Parameters Id.";
                        }
                        container yang-ext {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            container docs-load-balancing {
                leaf enable {
                    type boolean;
                    default false;
                }
                container general-grp-defaults {
                    description "This object provides the default load balancing parameters for
General Load
                           Balancing Groups (MD-CM-SGs) that are used when instances of
GeneralGrpCfg are
                           created by the CMTS.";
                    leaf enable {
                        type boolean;
                    }
                    leaf policy-id {
                        type leafref {
                            path "../../load-balancing-policy/policy-id";
                        }
                    }
                    leaf init-tech {
                        type load-bal-init-tech-type;
                        default "reinit-mac broadcast-ranging unicast-ranging ranging direct";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list load-balancing-policy {
                    key policy-id;
                    description "This object describes the set of load balancing policies. Instances
from
                           this object might be referenced by GrpStatus object. All the rules
contained
                           in a load balancing policy apply to an Autonomous Load Balancing
operations.
                           Load balancing rules are defined within the present document or can be
                           vendor-defined as well.";
                    leaf policy-id {
                        type uint32;
                    }
                    list load-balance-rule {
                        key rule-id;
                        min-elements 1;
                        leaf rule-id {
                            type leafref {
                                path "../../../basic-rule/rule-id";
                            }
                        }
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list basic-rule {
                    key rule-id;
```

```
                    description "This object represents a basic ruleset applicable to a load
balancing
                          policy that references it.";
                    leaf rule-id {
                        type uint32;
                    }
                    leaf enable {
                        type load-balancing-rule-type;
                        description "This attribute when set to 'enabled' enables Autonomous Load
                               Balancing (independently of the load balancing group enable/disable
state),
                               the rule set is disabled if set to 'disabled', or is disabled during
a
                               period of time it set to 'disabledPeriod'.";
                    }
                    leaf disable-start {
                        type uint32 {
                            range "0..86399";
                        }
                        description "This attribute disables load balancing from the time stated by
this
                               attribute when the attribute Enable is set to 'disablePeriod'. The
time is
                               defined in seconds since midnight.";
                    }
                    leaf disable-end {
                        type uint32 {
                            range "1..86400";
                        }
                        description "This attribute disables load balancing until the time stated by
this
                               attribute when the attribute Enable is set to 'disabled-period'. The
time is
                               defined in seconds of the wall clock since midnight. Is to be greater
than
                               disable-start.";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list general-grp-cfg {
                    key "mac-domain-name";
                    description "This object provides the parameters for the General Load Balancing
Groups
                          of MD-CM-SGs associated with MAC Domain-Fiber Node pairs. This object
allows
                          configuration of load balancing parameters for General Load Balancing
Groups by
                          way of MAC Domain-Fiber Node pairs.  In many deployments, a MAC Domain-
Fiber Node
                          pair will equate to an MD-CM-SG (which always equates to a GLBG).  In the
case
                          where an MD-CM-SG spans multiple Fiber Nodes, there will be multiple
elements in
                          the fiber node list.";
                    leaf mac-domain-name {
                        type leafref {
                            path "/ccap/docsis/docs-mac-domain/mac-domain/mac-domain-name";
                        }
                    }
                    list fiber-node {
                        key "fiber-node-index";
                        min-elements 1;
                        leaf fiber-node-index {
                            type leafref {
                                path "/ccap/chassis/fiber-node-config/fiber-node-config-index";
                            }
                        }
                        container yang-ext {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                    leaf policy-id {
                        type leafref {
                            path "../../load-balancing-policy/policy-id";
```

```
                        }
                        mandatory true;
                    }
                    leaf enable {
                        type boolean;
                        default true;
                    }
                    leaf init-tech {
                        type load-bal-init-tech-type;
                        default "";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list restricted-grp-cfg {
                    key res-grp-id;
                    description "This object represents the configuration of Restricted Load
Balancing Groups.";
                    leaf res-grp-id {
                        type uint32;
                    }
                    list grp-mac-domain {
                        key mac-domain-name;
                        min-elements 1;
                        leaf mac-domain-name {
                            type leafref {
                                path "/ccap/docsis/docs-mac-domain/mac-domain/mac-domain-name";
                            }
                            description "This attribute represents one or more MAC domain where the
Restricted Load
                                 balancing Group applies. Vendors are only required to support one
MAC Domain.";
                        }
                    }
                    leaf init-tech {
                        type load-bal-init-tech-type;
                    }
                    leaf policy-id {
                        type leafref {
                            path "../../load-balancing-policy/policy-id";
                        }
                    }
                    leaf service-type-id-list {
                        type snmp-tag-list-type;
                        description "This attribute represents the Service Type Id associated
                                     with this cable modem MAC address - MAC Address
                                     mask combination.. The zero-length string indicates
                                     that the instance is matched only against the GrpId
                                     value, if both GrpId and this attribute are not present
                                     the instance is ignored for matching purposes.";
                    }
                    list upstream-logical-channel-ref {
                        key "slot us-rf-port upstream-physical-channel upstream-logical-channel";
                        min-elements 0;
                        description "This element configures one or more upstream logical channel to
be
                                part of this restricted load-balancing group.";
                        uses upstream-logical-channel-reference;
                        container yang-ext {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                    list docsis-down-channel-ref {
                        key "slot ds-rf-port down-channel";
                        description "A reference to a DOCSIS downstream channel.";
                        ccap:inlineType;
                        uses docsis-down-channel-reference;
                        container yang-ext {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
```

```
                    }
                }
                list restrict-cm-cfg {
                    key restricted-cm-id;
                    description "This object describes the list of cable modems being statically
provisioned
                                 at the CMTS to a Restricted Load Balancing Group.";
                    leaf restricted-cm-id {
                        type uint32;
                    }
                    leaf mac-addr {
                        type yang:mac-address;
                        default 00:00:00:00:00:00;
                    }
                    leaf mac-mask {
                        type yang:mac-address;
                        default FF:FF:FF:FF:FF:FF;
                    }
                    leaf service-type-id {
                        type string {
                            length "0..16";
                        }
                    }
                    leaf restricted-grp-id {
                        type leafref {
                            path "../../restricted-grp-cfg/res-grp-id";
                        }
                        description "The attribute represents the Restricted Load Balancing Group
identifier
                                of this entry associated with the cable modem MAC address - MAC
address mask
                                combination. If not present, then the instance is matched only
                                against the ServiceTypeId value.";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            container yang-ext {
                ccap:extensionPoint; //different pyang flags impact use of this hint
                description "node for vendor YANG extensions";
            }
        }
        container video {
            description "The video object is the primary container of video configuration objects.";
            reference
                "CCAP Operations Support System Interface Specification
                 CM-SP-CCAP-OSSI-I01-110930 CCAP Video Session Configuration Objects section.";
            container global-input-ts-config {
                description "This object represents the global configuration of input transport
streams.";
                leaf jitter-tolerance {
                    type uint32;
                    units milliseconds;
                    default 100;
                    description "This attribute represents the acceptable delay variation in
millisecond for incoming streams. The jitter option sets the size of a dejittering buffer that
absorbs the input jitter of a session.";
                }
                leaf unicast-session-loss-timeout {
                    type uint32;
                    units milliseconds;
                    default 5000;
                    description "This attribute represents the loss of signal timeout in
milliseconds for unicast input streams. ";
                    reference "ANSI SCTE 154-4 2008, MPEG Management Information Base SCTE-HMS-MPEG-
MIB mpegLossOfSignalTimeout.";
                }
                leaf multicast-session-loss-timeout {
                    type uint32;
                    units milliseconds;
                    default 5000;
```

```
                    description "This attribute represents the loss of signal timeout in
milliseconds for the multicast input streams.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            container global-output-ts-config {
                description "This object represents the global configuration of output transport
streams.";
                leaf cat-insert-rate {
                    type uint8 {
                        range "0..32";
                    }
                    units "tables per second";
                    default 10;
                    description "This attribute represents the CAT insertion rate expressed in
tables/second.";
                    reference "ANSI SCTE 154-4 2008, MPEG Management Information Base SCTE-HMS-MPEG-
MIB mpegOutputTSCatInsertRate";
                }
                leaf pat-insert-rate {
                    type uint8 {
                        range "0..32";
                    }
                    units "tables per second";
                    default 10;
                    description "This attribute represents the PAT table interval expressed in
tables/second.";
                    reference "ANSI SCTE 154-4 2008, MPEG Management Information Base SCTE-HMS-MPEG-
MIB mpegOutputTSPatInsertRate";
                }
                leaf pmt-insert-rate {
                    type uint8 {
                        range "0..32";
                    }
                    units "tables per second";
                    default 10;
                    description "This attribute represents the PMT table interval expressed in
tables/second.";
                    reference "ANSI SCTE 154-4 2008, MPEG Management Information Base SCTE-HMS-MPEG-
MIB mpegOutputTSPatInsertRate";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list video-input-ts {
                key input-ts-index;
                description
                    "The VideoInputTs object configures a given MPEG-2 Transport stream that may be
unicast or multicast.";

                leaf input-ts-index {
                    type uint32;
                    description
                      "This is the index for an instance of the video-input-ts list";
                }
                leaf input-ts-name {
                    type string;
                    default " ";
                    description "This attribute configures a name for the input TS. ";
                }
                leaf input-ts-decryption-enabled {
                    type boolean;
                    default false;
                    description
                      "This attribute configures whether this input stream is encrypted for
transport across the WAN. This WAN encryption is intended to be removed at the CCAP and not related
to any CA encryption that may be configured for the output stream. A value of true means that the
CCAP needs to decrypt this input stream as applicable. A value of false means that the CCAP does not
need to decrypt this input stream.";
                }
                choice input-ts {
                    mandatory true;
                    case unicast-video-input-ts {
```

```
                            choice unicast-destination {
                                mandatory true;
                                case address {
                                    leaf unicast-ts-destination-ip-address {
                                        type inet:ip-address;
                                        mandatory true;
                                        description "This attribute corresponds to the IP
destination address of the UDP IP flow of the input TS.";
                                    }
                                }
                                case interface {
                                    leaf unicast-ts-interface-name{
                                        type string;
                                        mandatory true;
                                        description "Local IP interface; needs to be a configured
ip-interface name. This association provides a static mapping of an input transport stream to an IP
interface";
                                    }
                                }
                            }
                            leaf unicast-ts-destination-udp-port {
                                type inet:port-number;
                                mandatory true;
                                description "This attribute corresponds to the UDP destination port
of the UDP IP flow of the input TS.";
                            }
                            container yang-ext1 {
                                ccap:extensionPoint; //different pyang flags impact use of this hint
                                description "node for vendor YANG extensions";
                            }
                        }
                    case multicast-video-input-ts {
                            list multicast-ts {
                                key "multicast-ts-source-ip-address multicast-ts-destination-ip-
address";
                                min-elements 1;
                                max-elements 2;
                                description "This object specifies the multicast flows of an input
transport stream. Having two MulticastTsVideoInput objects for one InputTs occurs when input TS
redundancy is configured (Hot-Hot sparing). If two MulticastTsVideoInput objects have the same
Priority, this implies HOT-HOT redundancy. Which stream is actually forwarded is vendor-specific.";
                                leaf multicast-ts-source-ip-address {
                                    type inet:ip-address;
                                    mandatory true;
                                    description "This attribute corresponds to the Source Specific
Multicast IP Address of the UDP IP flow.";
                                }
                                leaf multicast-ts-destination-ip-address {
                                    type inet:ip-address;
                                    mandatory true;
                                    description "This attribute corresponds to the group address of
a SSM (Source Specific Multicast) origination input TS.";
                                }
                                leaf multicast-ts-destination-udp-port {
                                    type inet:port-number;
                                    default 0;
                                    description "This attribute corresponds to the UDP destination
port of the UDP IP flow of the input TS.";
                                }
                                leaf multicast-ts-priority {
                                    type int8;
                                    mandatory true;
                                    description "This attribute is a number that identifies the
preference order of this transport stream; higher number indicates a higher priority. It is used to
order the multicast transport stream for the purpose of redundancy in the case of multiple multicast
video sources. If two entries have the same 'Priority', it implies Hot-Hot redundancy.";
                                }
                                leaf multicast-ts-interface-name{
                                    type string;
                                    description "Local IP interface; needs to be a configured ip-
interface name. This association provides a static mapping of an input transport stream to an IP
interface";
                                }
                                container yang-ext {
                                    ccap:extensionPoint; //different pyang flags impact use of this
hint
                                    description "node for vendor YANG extensions";
                                }
```

```
                        }
                        container yang-ext2 {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                    case yang-choice-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list static-udp-map {
                key udp-map-index;
                min-elements 0;
                description "This object represents the UDP port ranges used for static video
sessions.";
                uses udp-map;
                leaf static-video-output-ts {
                    mandatory true;
                    type leafref {
                        path "/ccap/video/video-output-ts/output-ts-index";
                    }
                    description "Reference to video-output-ts index.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list reserved-udp-map {
                key udp-map-index;
                min-elements 0;
                description "This object represents reserved ports to be used for non-video
applications.";
                uses udp-map;
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list reserved-pid-range {
                key reserved-pid-range-index;
                min-elements 0;
                description "This object represents reserved PID range to not be used on ERM
assignments.";
                leaf reserved-pid-range-index {
                    type uint32;
                    mandatory true;
                    description "This key represents the unique identifier of an instance in this
object.";
                }
                leaf starting-pid {
                    type uint32;
                    default 0;
                    description "This attribute represents the PID range starts for other
applications reserved PIDs.";
                }
                leaf count {
                    type uint32;
                    default 0;
                    description "This attribute represents the number of reserved PIDs starting from
'StartingPid' attribute value.";
                }
                leaf description {
                    type string;
                    default "";
                    description "This attribute represents the description associated with a PID
range configured instance.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
```

```
            }
            list input-registration {
                key input-registration-name;
                min-elements 0;
                description "This object represents the configuration of Edge ERRP parameters.";
                leaf input-registration-name {
                    type string;
                    mandatory true;
                    description "This key represents the Input interface name. This name corresponds
to the [RFC 4133], ENTITY-MIB entPhysicalName.";
                    reference "RFC 4133 ENTITY-MIB entPhysicalName";
                }
                leaf group-name {
                    type string;
                    description "This attribute represents the name of the Edge Input Group
associated with this input. This parameter is used in the ERRP Edge Input attribute.";
                }
                leaf erm-name {
                    type string;
                    description "This attribute represents the ERM where the QAM channel is
advertised. If empty, the QAM channel is not advertised.";
                }
                leaf bandwidth {
                    type uint32;
                    default 0;
                    description "This attribute represents the bandwidth of the edge input to be
advertised. If zero or not present, the CCAP advertises the full bandwidth of the edge input. If the
attribute erm-managed-input is set to false, operators should set the bandwidth attribute to a value
that greatly exceeds the speed of the input interface; this will cause the ERM to not actively
manage the input bandwidth.";
                }
                leaf erm-managed-input {
                    type boolean;
                    mandatory true;
                    description "This attribute allows the Operator to configure whether or not the
ERM should manage the input bandwidth on this EdgeInput Interface. A value of true indicates that
the ERM will manage the input bandwidth; a value of false indicates that the CCAP will manage the
input bandwidth. If set to false, operators should set the bandwidth attribute to a value that
greatly exceeds the speed of the input interface so that the ERM will not actively manage the input
bandwidth.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list pid-session {
                key session-index;
                min-elements 0;
                description "The PidSession object allows the identification, processing and
insertion of a single PID stream into exactly one OutputTs. Each PidSession object needs to have one
InputTs object associated with it.";
                uses video-session-group;
                leaf input-pid {
                    type uint16 {
                        range "0..8191 | 65535";
                    }
                    mandatory true;
                    description "This attribute identifies a specific PID stream in the input
transport stream.";
                    reference "ANSI SCTE 154-5 2008, SCTE-HMS-HEADENDIDENT TEXTUAL CONVENTIONS MIB
HePidValue.";
                }
                leaf pid-remap-enable {
                    type boolean;
                    default false;
                    description "This object configures whether or not the identified PID stream can
be remapped when inserted in the OutputTs.";
                }
                leaf pid-type {
                    type video-pid-usage-type;
                    mandatory true;
                    description "This enumeration defines the type of the identified PID stream.
This value is used to understand what anchor table (i.e. PAT, CAT) would need to be updated, in case
PidRemapEnable is set to True and a remap is required.";


                }
```

```
                leaf cas-id {
                    type ccap-octet-data-type {
                       length "8";
                    }
                    default 00000000;
                    description "This attribute allows a proper identification of the CAT table
parameter(s) that need(s) to be updated when the PidType is set to 'EMM', PidRemapEnable is set to
True and a remap is required. This parameter is required in Simulcrypt operation, when the CAT
advertises more than one EMM PID streams. Zero means no CAS is associated with this PID session.";
                }
                leaf output-pid {
                    type uint16 {
                        range "0..8191 | 65535";
                    }
                    description "This attribute defines the expected PID value of the identified PID
stream when inserted in the OutputTS. However, the OutputPid value cannot be guaranteed if the pid-
remap-enable flag is set to True.";
                    reference "ANSI SCTE 154-5 2008, SCTE-HMS-HEADENDIDENT TEXTUAL CONVENTIONS MIB
HePidValue.";
                }
                leaf output-program-number {
                    type uint16;
                    description "This attribute identifies the output program number for this PID
Session.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list program-session {
                key session-index;
                min-elements 0;
                description "The ProgramSession object allows the identification, encryption,
processing and insertion a single program stream into exactly one OutputTs. Each ProgramSession
object needs to have one or two InputTs objects associated with it. Having two InputTs objects
associated with it occurs when input TS redundancy is configured (Hot-Hot sparing).";
                uses video-session-group;
                leaf input-mpeg-program-number {
                    type uint16;
                    mandatory true;
                    description "This attribute selects a specific program from the transport stream
of the incoming video stream. This program number should be part of the incoming PAT. A value of 0
(zero) means that any incoming program number can be accepted.";
                }
                leaf output-mpeg-program-number {
                    type uint16;
                    mandatory true;
                    description "This attribute specifies the program number to be present in the
transport stream of the outgoing video stream. This program number will be part of the outgoing
PAT.";
                }
                leaf pat-pid-remap {
                    type boolean;
                    default true;
                    description "A value true indicates that the elementary stream PID of this input
program can be remapped to the output TS, as long as the PAT and PMT are updated. A value false
indicates that the same input elementary stream PID has to be used on the output TS.";
                }
                leaf requested-bandwidth {
                    type uint32;
                    units bps;
                    mandatory true;
                    description "This attribute configures the expected bandwidth parameters for a
static input video session described by this object. This parameter is used for encryption and video
down channel output resources. A value of 0 (zero) means that no bandwidth validation is required.";
                }
                leaf cas-info {
                    type leafref {
                        path "/ccap/video/cas-info/cas-info-index";
                    }
                    description "Reference to a cas-info index.";
                }
                leaf encryption-data {
                    type leafref {
                        path "/ccap/video/encryption-data/encryption-data-index";
                    }
                    description "A reference to encryption-data index.";
```

```
                }
                leaf encrypt-control {
                    type leafref {
                        path "/ccap/video/encrypt-control/encrypt-control-index";
                    }
                    description "A reference to encrypt-control index.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list cas-info {
                key cas-info-index;
                min-elements 0;
                description
                    "The cas-info object serves two purposes:
                     1. It identifies the ECMG(s) that need(s) to be involved in the encryption of
the program session. In the case of a Simulcrypt operation, more than one CasInfo object can be
attached to the same ProgramSession.
                     2. It defines a CA-specific opaque object that needs to be forwarded to the
appropriate ECMG when the session is initialized. ";
                leaf cas-info-index {
                    type uint32;
                    mandatory true;
                    description "This attribute configures the index for an instance of CasInfo for
a given ProgramSession.";
                }
                leaf cas-id {
                    type ccap-octet-data-type {
                       length "8";
                    }
                    mandatory true;
                    description "CasId is the 32 bit hexadecimal representation of the CAS system
identifier.";
                }
                leaf ca-blob {
                    type string;
                    mandatory true;
                    description "CaBlob is opaque data that the Encryptor is required to forward to
the ECMG associated with the specified CasId.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list mpts-passthrough-session {
                key session-index;
                min-elements 0;
                description "The mpts-passthrough-session object allows the identification and
insertion of an unmodified MPTS into exactly one OutputTs. Each MptsPassThruSession object needs to
have one or two InputTs objects associated with it. Having two InputTs objects associated with it
occurs when input TS redundancy is configured (Hot-Hot sparing).";
                uses video-session-group;
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list encryption-data {
                key encryption-data-index;
                min-elements 0;
                description "The EncryptionData object allows a per video session encryption
configuration.";
                leaf encryption-data-index {
                    type uint32;
                    mandatory true;
                    description "The index is the key for the EncryptionData object.";
                }
                leaf cci-level {
                    type cci-level-type;
                    mandatory true;
                    description "This attribute represents the Copy Control Indicator/Digital Rights
protection applicable to the program. It is forwarded to all active ECMGs to be encapsulated into
ECMs.";
                }
                leaf cit {
```

```
                    type cit-type;
                    mandatory true;
                    description "This attribute represents the Constrained Image Trigger flag
applicable to the program. It is forwarded to all active ECMGs to be encapsulated into ECMs.";
                }
                leaf rct {
                    type rct-type;
                    mandatory true;
                    description "This attribute represents the Redistribution Control Trigger flag
applicable to the program. It is forwarded to all active ECMGs to be encapsulated into ECMs.";
                }
                leaf cci-reserved {
                    type cci-reserved-type;
                    mandatory true;
                    description "This attribute reserves 2 bits of copy control information (CCI)
for future use. It is forwarded to all active ECMGs to be encapsulated into ECMs.";
                }
                leaf provider-asset-id {
                    type string {
                        length "1..255";
                    }
                    mandatory true;
                    description "This attribute configures the Provide Asset Id parameter that is
passed in the initial RTSP session SETUP (e.g. for VOD) to the Encryptor and enables the Encryptor
to uniquely identify/reference the VOD asset or broadcast channel. ";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list encrypt-control {
                key encrypt-control-index;
                min-elements 0;
                description "This configuration object selects the encryption option of a static
encryption session. The CaEncryptionMode is not defined as only one mode and is supported per
Encryptor.";
                leaf encrypt-control-index {
                    type uint32;
                    mandatory true;
                    description "This attribute configures the index for an instance of
EncryptControl for a given ProgramSession.";
                }
                leaf encryption-scheme {
                    type encryption-scheme-type;
                    mandatory true;
                    description "This attribute defines the encryption algorithm to be used for a
given video session. The value of other(1) is used when a vendor-extension has been implemented for
this attribute.";
                }
                leaf block-stream-until-encrypted {
                    type boolean;
                    default true;
                    description "BlockStreamUntilEncrypted indicates if the encryption engine should
block the program until it can encrypt it (i.e. it has received a first Encryption Control Message
(ECM) and Control Word (CW) from the ECMG) or release it in the clear to the destination or output.
Values are 0 meaning false or 1 meaning true. Note that this parameter can be used to enforce
synchronous behavior, wherein the RTSP server (i.e. Encryption Engine) will not acknowledge the
session request back to the ERM until it has actually started to encrypt the stream. Obviously, this
assurance comes at the expense of setup latency.";
                }
                leaf key-length {
                    type key-length-type;
                    mandatory true;
                    description "This attribute configures the number of bits in the encryption keys
used by encryption algorithm defined by the EncryptionScheme attribute.";
                }
                leaf encryptor-opaque {
                    type string;
                    mandatory true;
                    description "EncryptorOpaque holds private data used by the Encryptor that may
be under CA license from the CA vendor.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
```

```
            list ecmd {
                key ecm-index;
                min-elements 0;
                description "This object allows for the configuration of the interface to an
Entitlement Control Message Decoder (ECMD). ";
                uses ecm-group;
                leaf number-decrypted-streams {
                    type uint32;
                    mandatory true;
                    description "The maximum number of decrypted streams the ECMD should handle.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list ecmg {
                key ecm-index;
                min-elements 0;
                description "This object allows for the configuration of the interface to an
Entitlement Control Message Generator (ECMG).  Redundant ECMGs for the same CAS may exist, each with
the same CA_System_ID, with the priority determining which is currently in use by an Encryptor for a
particular CAS. ";
                uses ecm-group;
                leaf recommended-cp-duration {
                    type uint32 {
                        range "1..4294967295";
                    }
                    units seconds;
                    mandatory true;
                    description "The recommended cryptoperiod, in seconds, the ECMG should expect.";
                }
                leaf number-encrypted-streams {
                    type uint32;
                    units streams;
                    mandatory true;
                    description "The maximum number of encrypted streams the ECMG should handle.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list erm-registration {
                key erm-name;
                min-elements 0;
                description "This object allows for the configuration of the interface to an Edge
Resource Manager. Only one configured ERM interface exists for the entire CCAP. An ErmRegistration
object contains the attributes in the following table. The CCAP may support only one instance of the
ErmRegistration object. Configuring more than one ERM is generally used for scaling purposes, with
each individual ERM being focused on specific, unique service groups. More than one ERM cannot be
practically used to support the same service group, and there could be conflicts between the control
messages of the independent ERMs.";
                leaf erm-name {
                    type string;
                    mandatory true;
                    description "This key represents the name of the ERM related to this
registration instance. This is an internal reference for associating, e.g. QAM channels and input
interfaces to an ERM.";
                }
                container erm-address {
                    uses host;
                    description "This attribute represents the IP address/hostname of the ERM. ";
                }
                leaf erm-port {
                    type inet:port-number;
                    default 0;
                    description "This attribute represents the TCP port number used to reach the
ERM.";
                }
                leaf erm-connection-mode {
                    type erm-connection-mode-type;
                    default client;
                    description "This attribute represents the type of TCP connection that is
established by the CCAP.";
                }
                leaf hold-timer {
                    type uint16 {
```

```
                    range "0 | 3..3600";
                }
                units seconds;
                default 240;
                description
                    "This attribute represents the number of seconds that the sender proposes
for the value of the hold timer. Upon receipt of an OPEN message, the CCAP shall calculate the value
of the Hold Timer by using the smaller of its configured Hold Time and the Hold Time received in the
OPEN message.
                     The Hold Time has to be either zero or at least three seconds.
                     An implementation may reject connections on the basis of the Hold Time. The
calculated value indicates the maximum number of seconds that may elapse between the receipt of
successive KEEPALIVE and/or UPDATE messages by the sender..";
            }
            leaf connection-retry-timer {
                type uint32;
                units seconds;
                default 20;
                description "This attribute represents the time in seconds for the connect retry
timer. The exact value of the connect retry timer is a local matter, but should be sufficiently
large to allow TCP initialization.";
            }
            leaf next-hop-address-domain {
                type uint32;
                mandatory true;
                description "This attribute represents the address domain number of the next-hop
server. The NextHopServer specifies the address to which a manager should use to connect to the
component in order to control the resource specified in the reachable route. This parameter is used
in the ERRP NextHopServer attribute.";
            }
            container comp-address {
                uses host;
                description "This attribute is the IP address/hostname of the ERRP
NextHopServer.";
            }
            leaf comp-port {
                type inet:port-number;
                default 6069;
                description "This attribute represents the ERRP NextHopServer attribute port
number. Only used if comp-address is configured.";
            }
            leaf streaming-zone {
                type string {
                    length "1..255";
                }
                mandatory true;
                description
                    "This attribute represents the Streaming Zone within which the component
operates. This parameter is used in the ERRP OPEN message. StreamingZone Name is a mandatory
parameter when supporting video applications. The capability is optional when signaling DOCSIS only
resources. Since the target for ERM configuration is video, this element is mandatory.
                     The value is to be set to the string that represents the StreamingZone
Name, i.e. <region>. The characters comprising the string are in the set within TEXT defined in
section 15.1 of [RFC 2326]. The CCAP shall support minimum string lengths of 64; however, the
composition of the string used is defined by implementation agreements specified by the service
provider.
                     A CCAP will exist in a single streaming zone.";
                reference "RFC 2326 Section 15.1";
            }
            leaf id {
                type uint32;
                default 0;
                description "This attribute represents the unique identifier for the CCAP device
within its Streaming Zone. This value can be set to the 4-octet value of an IPv4 address assigned to
that device. This ID value is determined on startup and is the same for all peer connections. This
parameter is used in the ERRP OPEN message header.";
            }
            leaf cost {
                type uint32;
                default 0;
                description "This attribute represents the static cost for use of this
resource.";
            }
            leaf comp-name {
                type string {
                    length "1..255";
                }
                mandatory true;
```

```
                    description
                        "The name of the component for which the data in the update message applies.
This parameter is used in the ERRP OPEN message. Component Name is a mandatory parameter when
supporting video applications. The capability is optional when signaling DOCSIS only resources.
Since the target for ERM configuration is video, this element is mandatory.
                         The value is to be set to the string that represents the Component Name,
i.e. <region>.<localname>. The characters comprising the string are in the set within TEXT defined
in section 15.1 of [RFC 2326]. The CCAP shall support minimum string lengths of 64; however, the
composition of the string used is defined by implementation agreements specified by the service
provider.";
                    reference "RFC 2326 Section 15.1";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list video-output-ts {
                key output-ts-index;
                min-elements 0;
                description "The video-output-ts element is an intermediate object that represents a
configuration multiplex of one or more ProgramSession instances.";
                leaf output-ts-index {
                    type uint32;
                    mandatory true;
                    description "This is not a tsid. It uniquely identifies a CCAP-generated MPTS
composed of one or more program streams, PID streams and/or pass thru MPTS.";
                }
                leaf output-ts-name {
                    type string;
                    default "";
                    description "This attribute configures the name of this instance of video-
output-ts.";
                }
                list video-down-channel-ref {
                    key "slot ds-rf-port down-channel";
                    min-elements 1;
                    description "Reference to a list of video down channels, all of which will
transmit this MPTS.";
                    uses video-down-channel-reference;
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list static-udp-map-encryption {
                key udp-map-encryption-index;
                leaf udp-map-encryption-index {
                    type uint8;
                }
                leaf cas-info {
                    mandatory true;
                    type leafref {
                        path "/ccap/video/cas-info/cas-info-index";
                    }
                    description "Reference to a cas-info index.";
                }
                leaf encryption-data {
                    type leafref {
                        path "/ccap/video/encryption-data/encryption-data-index";
                    }
                    description "A reference to encryption-data index.";
                }
                leaf encrypt-control {
                    mandatory true;
                    type leafref {
                        path "/ccap/video/encrypt-control/encrypt-control-index";
                    }
                    description "A reference to encrypt-control index.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
```

```
                }
            }
            container yang-ext {
                ccap:extensionPoint; //different pyang flags impact use of this hint
                description "node for vendor YANG extensions";
            }
        }
        container epon {
            description "The epon object is the primary container of EPON configuration objects.";
            container oam-config {
                description "This object controls the rate at which OAM messages are sent on the
EPON interface.";
                reference
                    "DOCSIS Provisioning of EPON OSSI Specification DPOE-SP-OSSIv1.0-I01-110225
                     EPON OAM Configuration section.";
                leaf min-oam-rate {
                    type uint8;
                    units "PDUs per second";
                    default 1;
                    description "The minimum number of OAM PDUs per second that will be generated by
the OLT. If no other OAM PDU is generated at the minimum rate, the OLT will generate an OAM
Information request as a 'heartbeat' message. A value of zero indicates that no heartbeat request
will be generated.";
                    reference
                        "DOCSIS Provisioning of EPON OSSI Specification DPOE-SP-OSSIv1.0-I01-110225
                         EPON OAM Configuration section.";
                }
                leaf max-oam-rate {
                    type uint8;
                    units "PDUs per second";
                    default 30;
                    description "The maximum number of OAM PDUs per second that can be exchanged
between the OLT and the ONU on a particular EPON link.";
                    reference
                        "DOCSIS Provisioning of EPON OSSI Specification DPOE-SP-OSSIv1.0-I01-110225
                         EPON OAM Configuration section.";
                }
                leaf oam-response-timeout {
                    type uint8;
                    units seconds;
                    default 1;
                    description "The maximum time (in seconds) that the OLT will wait for a response
from the ONU for a given OAM PDU request.";
                    reference
                        "DOCSIS Provisioning of EPON OSSI Specification DPOE-SP-OSSIv1.0-I01-110225
                         EPON OAM Configuration section.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            container loop-timing-config {
                description "This object configures the loop timing for EPON interfaces.";
                reference "DOCSIS Provisioning of EPON OSSI Specification DPOE-SP-OSSIv1.0-I01-
110225 Loop Timing section.";
                leaf min-propagation-delay {
                    type uint16;
                    units "16 ns TQ";
                    default 0;
                    description "The one way propagation delay (in 16ns units) on the fiber to the
closest ONU.";
                    reference "DOCSIS Provisioning of EPON OSSI Specification DPOE-SP-OSSIv1.0-I01-
110225 LoopTiming section.";
                }
                leaf max-propagation-delay {
                    type uint16;
                    units "16 ns TQ";
                    default 6250;
                    description "The one way propagation delay (in 16ns units) on the fiber to the
farthest ONU.";
                    reference "DOCSIS Provisioning of EPON OSSI Specification DPOE-SP-OSSIv1.0-I01-
110225 LoopTiming section.";
                }
                leaf onu-delay {
                    type uint16 {
                        range "3125..65535";
                    }
```

```
                    units "16 ns TQ";
                    default 3125;
                    description "The processing time (in 16ns units) required by an ONU.";
                    reference "DOCSIS Provisioning of EPON OSSI Specification DPOE-SP-OSSIv1.0-I01-
110225 LoopTiming section.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            container mpcp-config {
                description "This object configures the Multi-Point Control Protocol for EPON
interfaces.";
                reference "[DPoE OSSI], MPCP Configuration section";
                leaf discovery-period {
                    type uint16 {
                        range "10..65530";
                    }
                    units msec;
                    default 1000;
                    description "The period (in msecs) used by the OLT to generate discovery
gates.";
                    reference "DOCSIS Provisioning of EPON OSSI Specification DPOE-SP-OSSIv1.0-I01-
110225 MPCP Configuration section.";
                }
                leaf grant-size-in-discovery-gate {
                    type uint32 {
                        range "84..131070";
                    }
                    units Bytes;
                    default 16319;
                    description "The OLT's announced discovery grant length (in bytes).";
                    reference "DOCSIS Provisioning of EPON OSSI Specification DPOE-SP-OSSIv1.0-I01-
110225 MPCP Configuration section.";
                }
                leaf deregistration-timeout {
                    type uint32 {
                        range "0..4294967295";
                    }
                    units msec;
                    default 0;
                    description "The amount of time (in msecs) the OLT will wait for successive ONU
MPCP Report PDUs before deregistering the ONU.";
                    reference "DOCSIS Provisioning of EPON OSSI Specification DPOE-SP-OSSIv1.0-I01-
110225 MPCP Configuration section.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list deny-onu {
                key onu-mac-address;
                leaf onu-mac-address {
                    type yang:mac-address;
                    description "The MAC address of the ONU that will be added to the deny list.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list epon-mac-domain-config {
                key mac-domain-name;
                min-elements 0;
                description "This object defines a specialization of the MacDomain object for EPON
interfaces.";
                uses mac-domain-configuration-group;
                list one-gb-epon-interface {
                    key "epon-slot epon-port-number";
                    min-elements 0;
                    description "This configuration object allows for a one Gigabit EPON port to be
assigned to this epon-mac-domain. ";
                    uses epon-one-gb-port-reference;
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
```

```
                    }
                }
                list ten-gb-epon-interface {
                    key "epon-slot epon-port-number";
                    min-elements 0;
                    description "This configuration object allows for a symmetric or asymmetric ten
Gigabit EPON port to be assigned to this epon-mac-domain.";
                    uses epon-ten-gb-port-reference;
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            container yang-ext {
                ccap:extensionPoint; //different pyang flags impact use of this hint
                description "node for vendor YANG extensions";
            }
        }
        container network {
            description "Network is the primary container of network configuration objects.";
            list dns-resolver {
                key domain-suffix;
                min-elements 0;
                max-elements 1;
                description "This object allows the configuration of DNS servers and the
configuration of default domain suffix information.  The objects in this configuration object are
scalars.";
                leaf domain-suffix {
                    type string;
                    description "The attribute DomainSuffix configures a Domain Suffix that should
be post-pended to any hostname lookup that does not consist of a Fully Qualified Domain Name
(FQDN).";
                }
                leaf enabled {
                    type boolean;
                    default true;
                    description "This attribute configures if the associated domain suffix should be
applied to hostnames that do not include a Fully Qualified Domain Name (FQDN).";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list dns-server {
                key dns-server-index;
                min-elements 0;
                description "This object allows the configuration of the different DSN Servers that
the CCAP can use to get Domain Name Resolution.";
                leaf dns-server-index {
                    type uint32;
                    description "This attribute configures the index for this instance of
DnsServer.";
                }
                leaf server-ip {
                    type inet:ip-address;
                    mandatory true;
                    description "This attribute configures the IP address of the DNS server used by
the CCAP for DNS resolution. No distinction is made for IPv6 or IPv4 addresses here.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list integrated-servers {
                key "server-type local-listener-port";
                min-elements 0;
                description "This configuration object defines the types of servers integrated into
the CCAP and their respective administrative states.";
                leaf server-type {
                    type integrated-server-type;
                    mandatory true;
```

```
                    description "The type of server being configured on the CCAP.";
                }
                leaf local-listener-port {
                    type inet:port-number;
                    mandatory true;
                    description "The second part of the key for this object configures the TCP or
UDP port number on which the server listens. The CCAP shall assign the default value as the IANA-
assigned port number associated with the ServerType selected, as defined by IANA.";
                    reference "Port Numbers, IANA, http://www.iana.org/assignments/port-numbers";
                }
                leaf enabled {
                    type boolean;
                    default false;
                    description "This attribute configures the running state of the server.  True
means that the server will actively listen on the specified port.  False means that the specific
server is disabled.";
                }
                leaf listener-ip-interface-name {
                    type string;
                    description "Local IP interface; needs to be a configured ip-interface name.
This specifies the IP interface on which the server listens. If an IP interface is not specified,
the behavior of the CCAP is vendor specific.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            container ssh-server {
                description "This configuration object defines the types of servers integrated into
the CCAP and their respective administrative states.";
                leaf ssh-local-listener-port {
                    type inet:port-number;
                    default 22;
                    description "The second part of the key for this object configures the TCP or
UDP port number on which the server listens. The CCAP shall assign the default value as the IANA-
assigned port number associated with the ServerType selected, as defined by IANA.";
                    reference "Port Numbers, IANA, http://www.iana.org/assignments/port-numbers";
                }
                leaf enabled {
                    type boolean;
                    default false;
                    description "This attribute configures the running state of the server.  True
means that the server will actively listen on the specified port.  False means that the specific
server is disabled.";
                }
                leaf cipher {
                    type cipher-type;
                    default des3;
                    description "This attribute configures the set of encryption algorithms that
                                 are allowed on the SSH interface. SSH will use the enabled set
                                 of algorithms to negotiate the algorithm to use with the connecting
                                 client. The CCAP system shall log an error if the configuration
                                 file enables a cipher algorithm that is not supported. The bit
                                 setting of 'other' may be used to enable an algorithm supported
                                 by the CCAP that is not in the defined list.";
                }
                leaf message-auth-code {
                    type auth-code-type;
                    description "This attribute configures the set of encryption algorithms that
                                 are allowed on the SSH interface. SSH will use the enabled set of
                                 algorithms to negotiate the algorithm to use with the connecting
                                 client. The CCAP system shall log an error if the configuration
                                 file enables a cipher algorithm that is not supported. The bit
                                 setting of 'other' may be used to enable an algorithm supported
                                 by the CCAP that is not in the defined list.";
                }
                leaf host-authentication {
                    type host-auth-type;
                    default none;
                    description "This attribute enables SSH host authentication using public keys
                                 in a specified format. It is assumed that user authentication
                                 will be configured in the same way as other CCAP interfaces. The
                                 file format for key storage is outside the scope of the present
document.";
                }
                leaf listener-ip-interface-name {
                    type string;
```

```
                    description "Local IP interface; needs to be a configured ip-interface name.
This specifies the IP interface on which the server listens. If an IP interface is not specified,
the behavior of the CCAP is vendor specific.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            container telnet-server {
                description "This configuration object defines an integrated Telnet server in the
CCAP.";
                leaf telnet-local-listener-port {
                    type inet:port-number;
                    default 23;
                    description "The second part of the key for this object configures the TCP or
UDP port number on which the server listens. The CCAP shall assign the default value as the IANA-
assigned port number associated with the ServerType selected, as defined by IANA.";
                    reference "Port Numbers, IANA, http://www.iana.org/assignments/port-numbers";
                }
                leaf enabled {
                    type boolean;
                    default false;
                    description "This attribute configures the running state of the server.  True
means that the server will actively listen on the specified port.  False means that the specific
server is disabled.";
                }
                leaf listener-ip-interface-name {
                    type string;
                    description "Local IP interface; needs to be a configured ip-interface name.
This specifies the IP interface on which the server listens. If an IP interface is not specified,
the behavior of the CCAP is vendor specific.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list authentication-policy {
                key "protocol policy";
                min-elements 0;
                description "This configuration object allows the configuration of authentication
and accounting services. The Priority attribute controls which service is used first for
authenticating users.";
                leaf policy {
                    type policy-type;
                    description "This attribute is the first part of the key and configures the
policy type for the specified protocol.";
                }
                leaf protocol {
                    type protocol-type;
                    description "This attribute is the second part of the key and represents the
protocol used by the AAA server.";
                }
                leaf priority {
                    type uint32;
                    mandatory true;
                    description "This attribute sets a priority for the protocol selected. Higher
numbers are higher priority.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list local-authorization {
                key username;
                min-elements 0;
                description "Local-authorization configures the local user accounts and privilege
levels.";
                leaf username {
                    type string;
                    description "This attribute configures the login name to be used.";
                }
                leaf privilege-level {
                    type uint16 {
                        range "0..15";
                    }
```

```
                    mandatory true;
                    description "This attribute correspond to the user's privilege level. The
highest number provides the most user privileges.";
                }
                leaf password {
                    type string;
                    mandatory true;
                    description "This attribute correspond to the user's password. Upon export, the
CCAP shall export the Password attribute of the LocalAuth object encrypted with a vendor-specific
algorithm.";
                }
                leaf clear-key {
                    type boolean;
                    mandatory true;
                    description "This attribute indicates whether the Key/Password attribute is in
the clear (true) or encrypted (false). This attribute defines the status of the key/password
(encrypted or decrypted), not whether the device should export the key/password in the clear or
encrypted. Regardless of the value of this setting, the key/password will always be exported
encrypted. ";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list radius {
                key auth-server-index;
                min-elements 0;
                description "Radius creates the configuration for Radius servers.";
                uses authorizer-group;
                leaf radius-auth-port {
                    type inet:port-number;
                    default 1812;
                    description "This attribute defines the TCP port on which AAA authentication and
authorization are performed.";
                }
                leaf accounting-port {
                    type inet:port-number;
                    default 1813;
                    description "This attribute defines the TCP port on which AAA accounting is
performed.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list tacacs-plus {
                key auth-server-index;
                min-elements 0;
                description "Tacacs-plus configures TACACS+ services for the CCAP.";
                uses authorizer-group;
                leaf tacacs-plus-auth-port {
                    type inet:port-number;
                    default 49;
                    description "This attribute defines the TCP port used for communicating with the
AAA server.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list keychain {
                key key-id;
                min-elements 0;
                description "Keychain allows the CCAP to be configured with different Ripv2 key
change information.";
                leaf key-id {
                    type uint32 {
                        range "0..2147483647";
                    }
                    description "This attribute configures a KeyId used in RipV2 route updates.";
                }
                leaf key-string {
                    type string {
                        length "1..79";
                    }
```

```
                    mandatory true;
                    description "This attribute configures the actual key used for this instance.
This value has to be the same on both the sender and receiver of the RIPv2 route.";
                }
                leaf accept-lifetime {
                    type uint32 {
                        range "0..2147483647";
                    }
                    units seconds;
                    mandatory true;
                    description "This attribute configures the accept lifetime value in seconds for
the key in this instance.";
                }
                leaf send-lifetime {
                    type uint32 {
                        range "0..2147483647";
                    }
                    units seconds;
                    default 0;
                    description "This attribute configures the send lifetime value in seconds for
the key in this instance. A value of 0 (zero) means that there is no lifetime limit.";
                }
                leaf clear-key {
                    type boolean;
                    mandatory true;
                    description "This attribute indicates whether the key-string attribute is
included in the XML configuration file in the clear (true) or encrypted (false). This attribute
defines the status of the key (encrypted or decrypted), not whether the device should export the key
in the clear or encrypted. Regardless of the value of this setting, the key will always be exported
as encrypted.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            container fail-over {
                description "This object configures the automatic fail-over operation of the CCAP.";
                leaf auto-fail-back {
                    type boolean;
                    default true;
                    description "This attribute configures whether or not the CCAP automatically
switches back to a line card after a failover event. If true, when the failed card is operational,
the CCAP will begin using that card again. If False, the operator will have to perform the failback
operation.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            container local-time {
                description "Time allows the configuration of a Primary and Secondary NTP server, as
well as other local time attributes. This object does not fully configure all NTP client parameters.
Vendors may provide additional configuration objects to fully configure the NTP and SNTP protocols
if desired.";
                container ntp-master {
                    uses host;
                    description "This attribute represents the Master NTP server.";
                }
                container ntp-backup {
                    presence "Configures a backup NTP Server";
                    choice address-or-hostname {
                        case address {
                            leaf address {
                                mandatory true;
                                type inet:ip-address;
                            }
                        }
                        case name {
                            leaf name {
                                mandatory true;
                                type inet:domain-name;
                            }
                        }
                    }
                    description "This attribute represents the backup NTP Server in case the master
NTP fails.";
```

```
                }
                leaf time-zone {
                    type string;
                    default 00;
                    description
                        "This attribute represents the offset value to the local time to arrive at
UTC Time. The value has the following format:
                         hh[:mm] - the hour
                         (0 <= hh <= 24) - required, minutes
                         (0 <= mm <= 59) -the mm (minutes) is optional. The hour can be preceded by
a minus sign (-).";
                }
                leaf daylight-saving-time-shift-dates {
                    type string;
                    default "3.2.0/02:00,11.1.0/02:00,01";
                    description
                        "This attribute indicates when to change to and from daylight saving (or
summer) time.
                         The value has the form: date1/time1,date2/time2,offset.
                         The first date describes when the change from standard to daylight saving
time occurs, and the second date describes when the change back happens.
                         Each time field describes when, in current local time, the change to the
other time is made.
                         The format of date is the following: m.w.d - The dth day (0 <= d <= 6) of
week w of month m of the year (1 <= w <= 5, 1 <= m <= 12, where week 5 means 'the last d day in
month m', which may occur in the fourth or fifth week). Week 1 is the first week in which the dth
day occurs. Day zero is Sunday.
                         The time format is the following: hh:mm - The offset value is the value
that you add to the local time to arrive at UTC Time during the daylight saving time. The offset
value has the following format: hh[:mm].
                         The default value is the second Sunday in March (start) and the first
Sunday in November (end).

                         Reference: [PMI], Time Object section";
                }
                leaf dst-recurring-change {
                    type boolean;
                    description "This attribute controls whether the CCAP automatically adjusts the
time to Daylight Saving Time (DST). If true, the CCAP will adjust the time based on the value of the
DaylightSavingTimeCalendar attribute.";
                    default false;
                }
                leaf source-ip-interface-name {
                    type string;
                    description "Local IP interface; needs to be a configured ip-interface name. If
not specified, then the vendor picks the source IP address";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            container user-terminal
            {
                container console-terminal {
                    uses terminal-service-group;
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list virtual-terminal {
                    key vty-index;
                    leaf vty-index {
                        type uint8;
                    }
                    uses terminal-service-group;
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list acl
```

```
            {
                key acl-name;
                leaf acl-name {
                    type string;
                }
                list ip-acl-rule {
                    key acl-rule-index;
                    min-elements 1;
                    leaf acl-rule-index {
                        type uint16;
                    }
                    choice remark-or-action {
                        mandatory true;
                        case is-remark {
                            leaf remark { type string; mandatory true;}
                            container yang-ext1 {
                                ccap:extensionPoint; //different pyang flags impact use of this hint
                                description "node for vendor YANG extensions";
                            }
                        }
                        case is-rule {
                            leaf acl-action {
                                type authorization-action-type;
                                default deny;
                            }
                            choice address-family {
                                mandatory true;
                                case ipv4
                                {
                                    container ipv4-rule {
                                        description "This rule matches all of the defined elements.
                                                     If no filters are defined, it matches all IPv4
packets.";
                                        list dest-ipv4-addr-filter {
                                            max-elements 1;
                                            key dest-addr;
                                            leaf dest-addr {
                                                mandatory true;
                                                type inet:ipv4-address;
                                            }
                                            leaf dest-wildcard-mask {
                                                mandatory true;
                                                type inet:ipv4-address;
                                                                description "The dest-wildcard-mask
attribute defines which bits of the packet's
                                                                             source IP address are
matched to the dest-wildcard-mask attribute. The
                                                                             usage of the IP address
wildcard differs from most typical applications
                                                                             where IP addresses are
masked. Rather than restricting the defined IP
                                                                             address to a range of
addresses by masking off the lowest significant
                                                                             bits of the address,
the IP address mask is used as a wildcard.

                                                                             Each bit in the dest-
wildcard-mask set to zero indicates that the
                                                                             corresponding bit
position in the packet's source IP address needs to exactly
                                                                             match the bit value in
the corresponding bit position in the dest-wildcard-mask.
                                                                             Each wildcard bit set
to one indicates that both a zero bit and a one bit
                                                                             in the corresponding
position of the packet's IP address will be considered a
                                                                             match to this access
list entry. In other words, 'ones' are places in bit
                                                                             positions that should
be ignored. The set of 'ones' does not have to start at
                                                                             LSB, nor has to cover
consecutive bit positions. For example, a value of
                                                                             0.0.255.1 is valid for
an IPv4 wildcard.";
                                            }
                                        }
                                        list source-ipv4-addr-filter {
                                            max-elements 1;
```

```
                                        key source-addr;
                                        leaf source-addr {
                                            mandatory true;
                                            type inet:ipv4-address;
                                        }
                                        leaf source-wildcard-mask {
                                            mandatory true;
                                            type inet:ipv4-address;
                                                    description "The source-wildcard-
mask attribute defines which bits of the packet's
                                                                 source IP address are
matched to the source-wildcard-mask attribute. The
                                                                 usage of the IP address
wildcard differs from most typical applications
                                                                 where IP addresses are
masked. Rather than restricting the defined IP
                                                                 address to a range of
addresses by masking off the lowest significant
                                                                 bits of the address,
the IP address mask is used as a wildcard.

                                                                 Each bit in the source-
wildcard-mask set to zero indicates that the
                                                                 corresponding bit
position in the packet's source IP address needs to exactly
                                                                 match the bit value in
the corresponding bit position in the source-wildcard-mask.
                                                                 Each wildcard bit set
to one indicates that both a zero bit and a one bit
                                                                 in the corresponding
position of the packet's IP address will be considered a
                                                                 match to this access
list entry. In other words, 'ones' are places in bit
                                                                 positions that should
be ignored. The set of 'ones' does not have to start at
                                                                 LSB, nor has to cover
consecutive bit positions. For example, a value of
                                                                 0.0.255.1 is valid for
an IPv4 wildcard.";
                                        }
                                    }
                                    uses acl-filter-group;
                                    container yang-ext {
                                        ccap:extensionPoint; //different pyang flags impact use
of this hint
                                        description "node for vendor YANG extensions";
                                    }
                                }
                            }
                            case ipv6
                            {
                                container ipv6-rule
                                {
                                    description "This rule matches all of the defined elements.
                                                 If no filters are defined, it matches all IPv6
packets.";
                                    list dest-ipv6-addr-filter {
                                        max-elements 1;
                                        key dest-addr;
                                        leaf dest-addr {
                                            mandatory true;
                                            type inet:ipv6-address;
                                        }
                                        leaf dest-wildcard-mask {
                                            mandatory true;
                                            type inet:ipv6-address;
                                                    description "The dest-wildcard-mask
attribute defines which bits of the packet's
                                                                 source IP address are
matched to the dest-wildcard-mask attribute. The
                                                                 usage of the IP address
wildcard differs from most typical applications
                                                                 where IP addresses are
masked. Rather than restricting the defined IP
                                                                 address to a range of
addresses by masking off the lowest significant
                                                                 bits of the address,
the IP address mask is used as a wildcard.
```

```
                                                                    Each bit in the dest-
wildcard-mask set to zero indicates that the
                                                                    corresponding bit
position in the packet's source IP address needs to exactly
                                                                    match the bit value in
the corresponding bit position in the dest-wildcard-mask.
                                                                    Each wildcard bit set
to one indicates that both a zero bit and a one bit
                                                                    in the corresponding
position of the packet's IP address will be considered a
                                                                    match to this access
list entry. In other words, 'ones' are places in bit
                                                                    positions that should
be ignored. The set of 'ones' does not have to start at
                                                                    LSB, nor has to cover
consecutive bit positions. For example, a value of
                                                                    0.0.255.1 is valid for
an IPv4 wildcard.";
                                        }
                                    }
                                    list source-ipv6-addr-filter {
                                        max-elements 1;
                                        key source-addr;
                                        leaf source-addr {
                                            mandatory true;
                                            type inet:ipv6-address;
                                        }
                                        leaf source-wildcard-mask {
                                            mandatory true;
                                            type inet:ipv6-address;
                                                    description "The source-wildcard-
mask attribute defines which bits of the packet's
                                                                    source IP address are
matched to the source-wildcard-mask attribute. The
                                                                    usage of the IP address
wildcard differs from most typical applications
                                                                    where IP addresses are
masked. Rather than restricting the defined IP
                                                                    address to a range of
addresses by masking off the lowest significant
                                                                    bits of the address,
the IP address mask is used as a wildcard.

                                                                    Each bit in the source-
wildcard-mask set to zero indicates that the
                                                                    corresponding bit
position in the packet's source IP address needs to exactly
                                                                    match the bit value in
the corresponding bit position in the source-wildcard-mask.
                                                                    Each wildcard bit set
to one indicates that both a zero bit and a one bit
                                                                    in the corresponding
position of the packet's IP address will be considered a
                                                                    match to this access
list entry. In other words, 'ones' are places in bit
                                                                    positions that should
be ignored. The set of 'ones' does not have to start at
                                                                    LSB, nor has to cover
consecutive bit positions. For example, a value of
                                                                    0.0.255.1 is valid for
an IPv4 wildcard.";
                                        }
                                    }
                                    uses acl-filter-group;
                                    container yang-ext {
                                        ccap:extensionPoint; //different pyang flags impact use
of this hint
                                        description "node for vendor YANG extensions";
                                    }
                                }
                            }
                            case yang-choice-ext {
                                ccap:extensionPoint; //different pyang flags impact use of this
hint
                                description "node for vendor YANG extensions";
                            }
                        }
```

```
                        container yang-ext2 {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                    case yang-choice-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            container yang-ext {
                ccap:extensionPoint; //different pyang flags impact use of this hint
                description "node for vendor YANG extensions";
            }
        }
        container yang-ext {
            ccap:extensionPoint; //different pyang flags impact use of this hint
            description "node for vendor YANG extensions";
        }
    }
    container interface {
        description "The interface object is the primary container of Interface configuration
objects.";
        list cable-bundle {
            key interface-index;
            min-elements 0;
            description "A CableBundle is a compact way of assigning Layer 3 network addresses
to a set of Layer 2 interfaces.  This allows the bundled Layer 2 interfaces to share a common pool
of IPv4 Subnets or IPv6 prefixes so that these IP address resources can be efficiently used by the
CCAP.";
            uses virtual-interface-group;
            leaf dhcp-giaddr-primary {
                type inet:ipv4-address;
                description "This attribute configures how the DHCP relay agent populates the
GiAddr for relayed DHCP traffic. Is optional for IPv6-only bundles and required for bundles using
IPv4.";
            }
            list secondary-giaddr {
                key dhcp-giaddr-secondary;
                description "The secondary GIADDR addresses";
                leaf dhcp-giaddr-secondary {
                    type inet:ipv4-address;
                }
                leaf application {
                    type application-type;
                    default all;
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            choice mac-domain {
                case docs-md {
                    list docsis-mac-domain {
                        key docsis-mac-domain-name;
                        leaf docsis-mac-domain-name {
                            type leafref {
                                path "/ccap/docsis/docs-mac-domain/mac-domain/mac-domain-name";
                            }
                            description "Reference to a specific docsis mac-domain-configuration
list entry";
                        }
                        container yang-ext {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                }
                case epon-md {
                    list epon-mac-domain {
                        key epon-mac-domain-name;
                        leaf epon-mac-domain-name {
```

```
                                type leafref {
                                    path "/ccap/epon/epon-mac-domain-config/mac-domain-name";
                                }
                                description "Reference to a specific epon mac-domain-configuration
list entry";
                            }
                            container yang-ext {
                                ccap:extensionPoint; //different pyang flags impact use of this hint
                                description "node for vendor YANG extensions";
                            }
                        }
                    }
                    case yang-choice-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list cable-helper-config {
                    key cable-helper-config-index;
                    min-elements 0;
                    description "The cable-helper-config configuration object allows the operator to
configure different CableHelper addresses for DHCP Clients.  These CableHelper addresses are tied to
the CableBundle interfaces and the MAC Domains they service.";
                    leaf cable-helper-config-index {
                        type uint32;
                        mandatory true;
                        description "The index for the CableHelperCfg instance.";
                    }
                    container cable-helper-address {
                        uses host;
                        description "This attribute configures the IP address/hostname of the DHCP
server configured as a cable helper.";
                    }
                    leaf application {
                        type application-type;
                        default all;
                        description "This attribute configures the device class for which this cable
helper configuration applies.";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                leaf ingress-acl {
                    type leafref {
                        path "/ccap/network/acl/acl-name";
                    }
                }
                leaf egress-acl {
                    type leafref {
                        path "/ccap/network/acl/acl-name";
                    }
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            list loopback {
                key interface-index;
                min-elements 0;
                description "A loopback interface is a logical interface that is not tied to a
specific hardware port.  It provides a virtual interface to assist in overall system configuration.
";
                uses virtual-interface-group;
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            container mgmd-router-interface {
                description "This configuration object allows for configuration of the CCAP IP
Multicast Router.";
                reference "RFC 5519 Multicast Group Membership Discovery MIB";
                leaf query-interval {
                    type uint32;
                    units seconds;
```

```
                    default 125;
                    description "The frequency in seconds at which IGMP or MLD Host-Query packets
are transmitted on this interface.";
                }
                leaf version {
                    type igmp-version-type;
                    default igmp-v2-or-mld-v1;
                    description "The version of MGMD that is running on this interface. This object
can be used to configure a router capable of running either version. For IGMP and MLD to function
correctly, all routers on a LAN need to be configured to run the same version on that LAN";
                }
                leaf query-max-response-time {
                    type uint32 {
                        range "0..31744";
                    }
                    units "tenths of seconds";
                    default 100;
                    description "The maximum query response interval in seconds advertised in MGMDv2
or IGMPv3 queries on this interface.";
                }
                leaf robustness {
                    type uint32 {
                        range "1..225";
                    }
                    default 2;
                    description "The robustness variable utilized by an MGMDv3 host in sending
state-change reports for multicast routers. To ensure the state-change report is not missed, the
host retransmits the state-change report [mgmdHostInterfaceVersion3Robustness - 1] times. The
variable needs to be a non-zero value.";
                }
                leaf last-member-query-interval {
                    type uint32 {
                        range "0..31744";
                    }
                    units "tenths of seconds";
                    default 10;
                    description "The Last Member Query Interval is the Max Query Response Interval
in tenths of a second inserted into group-specific queries sent in response to leave group messages,
and is also the amount of time between group-specific query messages. This value may be tuned to
modify the leave latency of the network. A reduced value results in reduced time to detect the loss
of the last member of a group. The value of this object is irrelevant if mgmdRouterInterfaceVersion
is 1.";
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            container yang-ext {
                ccap:extensionPoint; //different pyang flags impact use of this hint
                description "node for vendor YANG extensions";
            }
        }
        container management {
            description "The management object is the primary container of Management configuration
objects.";
            container ipdr {
                description "The ipdr object is the container for the IPDR configuration objects.";
                container exporter-config {
                    description "Exporter-config allows an IPDR Exporter to be turned on and off.";
                    leaf enabled {
                        type boolean;
                        default true;
                        description "This attribute configures whether or not the IPDR Exporter is
enabled.";
                    }
                    leaf ipdr-exporter-listener-interface-name {
                        type string;
                        description "Local IP interface; needs to be a configured ip-interface name.
This specifies the IP interface on which the IPDR server listens. If an IP interface is not
specified, the behavior of the CCAP is vendor specific.";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list streaming-session {
```

```
                    key session-id;
                    min-elements 0;
                    description "This configuration object is used to configure global IPDR
connection attributes. A typical use case is for a single Template to be associated with a
StreamingSession.";
                    leaf session-id {
                        type uint16;
                        mandatory true;
                        description "This attribute configures the ID for this instance.";
                    }
                    leaf keep-alive-interval {
                        type uint16;
                        units seconds;
                        default 20;
                        description "This attribute configures the interval in seconds at which IPDR
'keepalives' are sent from the CCAP IPDR exporter to the collector.";
                    }
                    leaf ack-time-interval {
                        type uint8 {
                            range "1..60";
                        }
                        units seconds;
                        default 30;
                        description "This attribute configures the interval in seconds in which the
CCAP IPDR exporter waits for an acknowledgment.";
                    }
                    leaf ack-sequence-interval {
                        type uint16 {
                            range "1..500";
                        }
                        units records;
                        default 200;
                        description "This attribute configures the maximum number of unacknowledged
records that can be sent by the CCAP IPDR exporter before receiving an acknowledgement.";
                    }
                    leaf collection-interval {
                        type uint32 {
                            range "0..86400";
                        }
                        units seconds;
                        mandatory true;
                        description
                            "Where streaming is of the type timeInterval, this attribute configures
the interval in seconds at which IPDR information is extracted from the CCAP management objects and
transmitted to the collector.
                             Where streaming is of the type timeEvent, this attribute identifies the
interval at which the CCAP IPDR exporter will close the IPDR session to allow IPDR session
processing to occur. Records created by Service Definitions supporting timeEvent are sent when the
event is generated.";
                    }
                    leaf streaming-type {
                        type ipdr-streaming-type;
                        mandatory true;
                        description "This attribute configures the type of IPDR streaming used for
the session.";
                    }
                    leaf enabled {
                        type boolean;
                        default true;
                        description "This attribute controls whether the IPDR Session is enabled or
disabled.";
                    }
                    list service-definition-template {
                        key service-definition-id;
                        min-elements 1;
                        leaf service-definition-id {
                            type ipdr-service-definition-type;
                            description "This list configures the service type definition(s) for
this IPDR session. See the IPDR Service Definition Schemas section of [OSSIv3.0] for the definitions
and schemas of the types defined in this enumeration. The value of other(1) is used when a vendor-
extension has been implemented for this attribute.";
                        }
                        container yang-ext {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                    list collector-reference {
```

```
                        key collector-id;
                        leaf collector-id {
                            type leafref {
                                path "../../../collector/collector-id";
                            }
                            description "A reference to a collector id.";
                        }
                        container yang-ext {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list collector {
                    key collector-id;
                    min-elements 0;
                    description "This configuration object allows the operator to configure an IPDR
collector.";
                    leaf collector-id {
                        type uint8;
                        mandatory true;
                        description "This key configures a unique identifier for this collector
instance.";
                    }
                    leaf collector-ip {
                        type inet:ip-address;
                        mandatory true;
                        description "This attribute configures the IP address of collectors from
which the CCAP will accept a connect. As per [OSSIv3.0], the collector establishes a connection to
the CCAP";
                    }
                    leaf collector-name {
                        type string;
                        default "";
                        description "This attribute configures a name for the IPDR collector. ";
                    }
                    leaf collector-port {
                        type inet:port-number;
                        default 4737;
                        description "This attribute configures the port used by the collector to
communicate with the CCAP.";
                    }
                    leaf priority {
                        type uint8;
                        mandatory true;
                        description "This attribute configures the priority of this IPDR collector.
The priority is used to elect the primary and active collector. The collector with the lowest
priority is elected.";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            container fault-management {
                description "The fault-management object is the primary container of fault
management configuration objects. ";
                container event-throttle-config {
                    description "This configuration object is based on the docsDevEvent object
defined in [RFC 4639] and will be used without modification for CCAP.
";
                    reference "RFC 4639 docsDevEvent Group";
                    leaf throttle-admin-state {
                        type event-throttle-admin-state-type;
                        default unconstrained;
                        description "Refer to RFC 4639.";
                    }
                    leaf threshold {
                        type uint32;
```

```
                        units events;
                        default 0;
                        description "Refer to RFC 4639.";
                        reference "RFC 4639";
                    }
                    leaf interval {
                        type uint32 {
                            range "1..2147483647";
                        }
                        units seconds;
                        default 1;
                        description "Refer to RFC 4639.";
                        reference "RFC 4639";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list event-reporting-config {
                    key priority;
                    min-elements 0;
                    description "This configuration object is based on the docsDevEvControlTable
object defined in [RFC 4639] and will be used without modification for CCAP.

";
                    reference "RFC 4639 docsDevEvControlTable";
                    leaf priority {
                        type syslog-priority-type;
                        mandatory true;
                        description "See RFC 4639 -- docsDevEvPriority";
                    }
                    leaf reporting {
                        type syslog-reporting-type;
                        mandatory true;
                        description "See RFC 4639 -- docsDevEvReporting";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list cmts-event-ctrl {
                    key event-id;
                    min-elements 0;
                    description
                        "This configuration object is defined in [OSSIv3.0] and will be used with no
modifications for CCAP.
                         This object represents the control mechanism to enable the dispatching of
events based on the Event Id. The following rules define the event control behavior:
                         - If the CmtsEventCtrl object has no instances or contains an instance with
Event ID 0, then all events matching the Local Log settings of docsDevEvReporting are sent to local
log ONLY.
                         - Additionally, if The CmtsEventCtrl object contains configured instances,
then Events matching the Event Ids configured in the object are sent according to the settings of
the docsDevEvReporting object; i.e. Traps, Syslog, etc. ";
                    reference
                        "DOCSIS 3.0 Operations Support System Interface Specification
                         CM-SP-OSSIv3.0-I15-110623 CmtsEventCtrl Object section.";
                    leaf event-id {
                        type uint32;
                        description "This attribute represents the Event ID of the event being
enabled to delivery to a dispatch mechanism (e.g. syslog).";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                choice snmp-trap-enable {
                    case trap-enable {
                        description "This configuration object contains attributes which allow
enabling or disabling of SNMP Notifications. The SnmpEnableAuthenTraps [RFC 3418] attribute will be
used without modification for the CCAP.


    ";
                        reference "RFC 3418 snmpEnableAuthenTraps";
```

```
                        leaf snmp-enable-authen-traps {
                            type boolean;
                            mandatory true;
                            description "See RFC 3418";
                            reference "RFC 3418";
                        }
                        container trap-enable-yang-ext {
                            ccap:extensionPoint; //different pyang flags impact use of this hint
                            description "node for vendor YANG extensions";
                        }
                    }
                }
                list interface-trap-enable {
                    key if-name;
                    min-elements 0;
                    description
                        "This configuration object is based on the ifLinkUpDownTrapEnable object
defined in [RFC 2863] and will be used without modifications for CCAP.
                         The LinkUpDownTrapEnable attribute is configurable per ifIndex from the
ifTable, as described in [OSSIv3.0].";
                    reference "[RFC 2863] ifLinkUpDownTrapEnable
";
                    leaf if-name {
                        type string {
                            length "0..255";
                        }
                        description "The path name of the interface.";
                        reference "RFC 2863 ifName";
                    }
                    leaf link-up-down-trap-enable {
                        type boolean;
                        mandatory true;
                        description "See RFC 2863";
                        reference "RFC 2863 ifLinkUpDownTrapEnable";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list syslog-server-config {
                    key syslog-server-config-index;
                    min-elements 0;
                    description "This object allows the configuration of a specific Syslog Server.";
                    leaf syslog-server-config-index {
                        type uint32;
                        description "This key represents the unique identifier of an instance in
this object.";
                    }
                    container syslog-server {
                        uses host;
                        description "This attribute represents the IP address/hostname of the syslog
server.";
                    }
                    leaf enabled {
                        type boolean;
                        default false;
                        description "Indicates if the syslog server is used for sending syslog
messages or is disabled.";
                    }
                    leaf syslog-source-ip-interface-name {
                        type string;
                        description "Local IP interface; needs to be a configured ip-interface name.
If not specified, then the vendor picks the source IP address";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                container diag-log-triggers-config {
                    description
                        "This configuration object is based on the LogTriggersCfg object in Annex G
of [OSSIv3.0] and will be used without modification for CCAP.
                         This object defines the parameters to configure the Diagnostic Log
triggers. One or more triggers can be configured to define the actions of creating or updating CM
entries into the Diagnostic Log.";
                    reference
```

```
                        "DOCSIS 3.0 Operations Support System Interface Specification
                         CM-SP-OSSIv3.0-I15-110623 LogTriggersCfg Object section.";
                    leaf include-triggers {
                        type trigger-flag-type;
                        default registration;
                        description "This attribute turns individual diagnostic triggers on and off
at a given time when each trigger is set to '1' or '0' respectively.";
                    }
                    leaf enable-aging-triggers {
                        type trigger-flag-type;
                        default " ";
                        description "This attribute enables and disables the aging of individual
triggers at a given time when each trigger is set to '1' or '0 respectively.  If a log entry is
added by multiple triggers, and aging is disabled for one of those triggers, the CMTS shall not age
out such entry.";
                    }
                    leaf reg-time-interval {
                        type uint32 {
                            range "60..86400";
                        }
                        units seconds;
                        default 90;
                        description "This attribute is an operator empirically derived, worst-case
number of seconds which the CM requires to complete registration. If the CM has not completed the
registration stage within this registration time interval, the CM will be added to the Diagnostic
Log.";
                    }
                    leaf reg-detail {
                        type reg-detail-type;
                        default " ";
                        description "Setting a bit representing a CM registration state will enable
counting the number of times the CMTS determines that such CM reaches that state as the last state
before failing to proceed further in the registration process and within the time interval
considered for the CM registration trigger detection.";
                    }
                    leaf ranging-retry-trigger {
                        type ranging-retry-trigger-type;
                        default consecutive-miss;
                        description "This attribute selects the type of ranging retry trigger to be
enable in the Diagnostic Log.";
                    }
                    leaf ranging-retry-threshold {
                        type uint8 {
                            range "3..12";
                        }
                        default 6;
                        description "This attribute indicates the number of times the CMTS does not
detect a CM acknowledgement of a MAC-layer station maintenance message from a CMTS to be exceeded in
order to add the CM to the Diagnostic Log. The value of RangingRetryType decides if consecutive
ranging miss or ranging miss ratio is used as trigger.";
                    }
                    leaf ranging-retry-station-maint-num {
                        type uint16 {
                            range "60..65535";
                        }
                        default 90;
                        description "This attribute indicates the number of station maintenance
opportunities to monitor for ranging retry trigger. This value implies time intervals in a certain
range. DOCSIS specifies that the CMTS schedules ranging opportunities to CMs sufficiently smaller
than T4. There is no fixed formula to derive at a fixed time interval, how many ranging
opportunities may be offered to a CM by the CMTS, hence using the number of station maintenance
opportunities provides ratio with the fixed denominators while also taking time factor into
consideration.";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                container diag-log-global-config {
                    description "This object defines the parameters to manage and control the
instantiation of CMs in the Diagnostic Log object.";
                    leaf max-size {
                        type uint32 {
                            range "1..4294967295";
                        }
                        units instances;
                        default 100;
```

```
                        description "This attribute indicates the maximum number of CM instances
that can be reported in the DiagLog.";
                    }
                    leaf notify-log-size-high-thrshld {
                        type uint32 {
                            range "1..4294967295";
                        }
                        units instances;
                        default 80;
                        description "This attribute is the high threshold value to send a
HighThreshold notification when the number of instances in the DiagLog exceeds this value.";
                    }
                    leaf notify-log-size-low-thrshld {
                        type uint32 {
                            range "1..4294967295";
                        }
                        units instances;
                        default 60;
                        description "This attribute is the threshold value to send a  LowThreshold
notification when the number of instances in DiagLog drops to this value, but only if the DiagLog
number of instances exceeded the NotifyLogSizeHighThrshld value.";
                    }
                    leaf aging {
                        type uint32 {
                            range "15..86400";
                        }
                        units minutes;
                        default 10080;
                        description "This attribute defines a period of time after which an instance
in the DiagLog and its corresponding  DiagLogDetail  instance (if present)  are removed if no longer
the DiagLog instance is updated due to a trigger detection process.";
                    }
                    leaf notif-ctrl {
                        type notif-ctrl-type;
                        default " ";
                        description "This attribute is used to enable diagnostic log related
notifications.";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            container snmp {
                description "This object defines the configuration of access control for SNMPv1/v2c
received request messages. When a SNMP request message is received, the system checks the validity
of the request by matching the community string, source (IP address, subnet),  access type and view
restrictions  for included SNMP OIDs in the request.";
                list access-config {
                    key community;
                    min-elements 0;
                    description "This object defines the configuration of access control for
SNMPv1/v2c received request messages. When a SNMP request message is received, the system checks the
validity of the request by matching the community string, source (IP address, subnet),  access type
and view restrictions  for included SNMP OIDs in the request.";
                    leaf community {
                        type string {
                            length "1..32";
                        }
                        mandatory true;
                        description "The community string defined for the access control rule.";
                    }
                    leaf ip-address {
                        type inet:ip-prefix;
                        mandatory true;
                        description "The prefix used to validate the source of an incoming SNMP
request.";
                    }
                    leaf type {
                        type snmp-access-type;
                        default read-only;
                        description "Defines the type of access granted to the SNMP request.";
                    }
```

```
                    uses view-config-ref;
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list view-config {
                    key view-name;
                    min-elements 0;
                    description "This object defines a View consisting of a single OID subtree
matching rule for inclusion or exclusion as part of a SNMP message processing procedure such as
access authorization or dispatch or notifications";
                    leaf view-name {
                        type string {
                            length "1..32";
                        }
                        mandatory true;
                        description "The administrative name of an instance of this object.";
                    }
                    leaf subtree {
                        type string;
                        mandatory true;
                        description
                            "The OID subtree to be matched for the access view. This attribute is
formatted as the text representation of an ASN.1 OID following the ABNF notation below:
                             Subtree =  empty | OID [.OID]*
                             OID = number ; 0..128
                             The matching procedures are borrowed from [RFC 3414] for tree views
matching with the difference that the configuration elements uses a text notation to represent OIDs
and OID masks. See the subtree-mask attribute definition for further information.";
                    }
                    leaf subtree-mask {
                        type string;
                        mandatory true;
                        description
                            "A mask to match OIDs for inclusion or exclusion as part of the view.
This attribute definition is borrowed from [RFC 3414]. The only difference is that instead of bits
per OID, a byte of value '0' or '1' is used to represent this attribute.
                             Each byte value 1 indicates the inclusion of the corresponding OID
position in the Subtree attribute, while the value 0 indicates no need to match. See [RFC 3414] for
details.";
                    }
                    leaf type {
                        type subtree-view-inclusion-type;
                        mandatory true;
                        description "Indicates inclusion or exclusion of the subtree for the defined
view.";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                list notification-receiver-config {
                    key notification-receiver-name;
                    min-elements 0;
                    description "This object defines where to send notifications. When an event is
to be dispatched as a notification, the system checks for instances of this object that have the
notification OID associated with the event as part of their Inclusion list in their view-
configuration instances. The system then sends notifications based on the matched occurrences per
their configured parameters.";
                    leaf notification-receiver-name {
                        type string {
                            length "1..32";
                        }
                        mandatory true;
                        description "The administrative name of an instance in this object.";
                    }
                    leaf type {
                        type notification-type;
                        default snmpv2c-trap;
                        description "Indicates the type of SNMP notification being sent.";
                    }
                    container notification-receiver {
                        uses host;
                        description "The IP address/hostname of the notification receiver.";
                    }
                    leaf notification-receiver-port {
```

```
                        type inet:port-number;
                        default 162;
                        description "The UDP port that the notification receiver listens on for
messages.";
                    }
                    leaf timeout {
                        type uint8;
                        units seconds;
                        default 1;
                        description "The time in seconds the sender waits for receiving confirmation
for a notification being sent. This attribute is meaningful only when the attribute type is set to
'snmpv2c-inform'; otherwise it is ignored.";
                    }
                    leaf retries {
                        type uint8;
                        default 3;
                        description "The number of retries the sender will attempt in case of it has
not received confirmation of inform reception. This attribute is meaningful only when the attribute
type is set to 'snmpv2c-inform'; otherwise it is ignored.";
                    }
                    uses view-config-ref;
                    leaf notif-source-ip-interface-name {
                        type string;
                        description "Local IP interface; needs to be a configured ip-interface name.
If not specified, then the vendor picks the source IP address";
                    }
                    container yang-ext {
                        ccap:extensionPoint; //different pyang flags impact use of this hint
                        description "node for vendor YANG extensions";
                    }
                }
                container yang-ext {
                    ccap:extensionPoint; //different pyang flags impact use of this hint
                    description "node for vendor YANG extensions";
                }
            }
            container yang-ext {
                ccap:extensionPoint; //different pyang flags impact use of this hint
                description "node for vendor YANG extensions";
            }
        }
        container yang-ext {
            ccap:extensionPoint; //different pyang flags impact use of this hint
            description "node for vendor YANG extensions";
        }
    }
    container ccap {
        description "The CCAP object serves as the root of the CCAP configuration data.  It consists
of three attributes that together describe the CCAP platform. ";
        uses ccap-group;
    }
}
```

Annex H (informative): Sample CCAP XML Configuration

H.1 CCAP XML Configuration File

```
<ccap:ccap xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" SchemaVersion="2013-04-04"
xsi:schemaLocation="urn:cablelabs:params:xml:ns:yang:ccap ccap@2013-04-04.xsd"
xmlns:ccap="urn:cablelabs:params:xml:ns:yang:ccap">
  <name>CCAP1</name>
  <description>Vendor A CCAP</description>
  <location>Denver</location>
  <vendor-extension-version>
    <major-version>1</major-version>
    <minor-version>0</minor-version>
  </vendor-extension-version>
  <chassis>
    <decryptor>
      <decryptor-index>1</decryptor-index>
      <cw-timeout>10</cw-timeout>
      <ecmd-usage>
        <ecmd-usage-index>1</ecmd-usage-index>
        <priority>1</priority>
        <ecmd-ref>1</ecmd-ref>
      </ecmd-usage>
    </decryptor>
    <fiber-node-config>
      <fiber-node-config-index>1</fiber-node-config-index>
      <fiber-node-name>Fiber Node 1</fiber-node-name>
      <ds-rf-port-ref>
        <slot>1</slot>
        <ds-rf-port>0</ds-rf-port>
      </ds-rf-port-ref>
      <us-rf-port-ref>
        <slot>9</slot>
        <us-rf-port>0</us-rf-port>
      </us-rf-port-ref>
    </fiber-node-config>
    <fiber-node-config>
      <fiber-node-config-index>2</fiber-node-config-index>
      <fiber-node-name>Fiber Node 2</fiber-node-name>
      <ds-rf-port-ref>
        <slot>1</slot>
        <ds-rf-port>1</ds-rf-port>
      </ds-rf-port-ref>
      <us-rf-port-ref>
        <slot>9</slot>
        <us-rf-port>1</us-rf-port>
      </us-rf-port-ref>
    </fiber-node-config>
    <fiber-node-config>
      <fiber-node-config-index>8</fiber-node-config-index>
      <fiber-node-name>Fiber Node 8</fiber-node-name>
      <ds-rf-port-ref>
        <slot>1</slot>
        <ds-rf-port>7</ds-rf-port>
      </ds-rf-port-ref>
      <us-rf-port-ref>
        <slot>9</slot>
        <us-rf-port>7</us-rf-port>
      </us-rf-port-ref>
    </fiber-node-config>
    <fiber-node-config>
      <fiber-node-config-index>9</fiber-node-config-index>
      <fiber-node-name>Fiber Node 9</fiber-node-name>
      <ds-rf-port-ref>
        <slot>3</slot>
        <ds-rf-port>0</ds-rf-port>
      </ds-rf-port-ref>
      <us-rf-port-ref>
        <slot>11</slot>
        <us-rf-port>0</us-rf-port>
```

```
  </us-rf-port-ref>
</fiber-node-config>
<fiber-node-config>
  <fiber-node-config-index>10</fiber-node-config-index>
  <fiber-node-name>Fiber Node10</fiber-node-name>
  <ds-rf-port-ref>
    <slot>3</slot>
    <ds-rf-port>1</ds-rf-port>
  </ds-rf-port-ref>
  <us-rf-port-ref>
    <slot>11</slot>
    <us-rf-port>1</us-rf-port>
  </us-rf-port-ref>
</fiber-node-config>
<fiber-node-config>
  <fiber-node-config-index>16</fiber-node-config-index>
  <fiber-node-name>Fiber Node 16</fiber-node-name>
  <ds-rf-port-ref>
    <slot>3</slot>
    <ds-rf-port>7</ds-rf-port>
  </ds-rf-port-ref>
  <us-rf-port-ref>
    <slot>11</slot>
    <us-rf-port>7</us-rf-port>
  </us-rf-port-ref>
</fiber-node-config>
<slot>
  <slot-number>1</slot-number>
   <rf-line-card>
    <rf-card>
     <line-card-name>Downstream RF Line Card 1</line-card-name>
     <admin-state>up</admin-state>
     <protected-by>2</protected-by>
    </rf-card>
    <encryptor>
      <encryptor-index>1</encryptor-index>
      <ca-encryptor-type>motorola</ca-encryptor-type>
      <ecm-timeout>10</ecm-timeout>
      <clear-stream-timeout>10</clear-stream-timeout>
      <ecmg-usage>
        <ecmg-usage-index>1</ecmg-usage-index>
        <priority>1</priority>
        <ecmg-ref>1</ecmg-ref>
      </ecmg-usage>
    </encryptor>
    <enable-udp-map-encryption>2</enable-udp-map-encryption>
    <ds-rf-port>
      <port-number>0</port-number>
      <rf-mute>false</rf-mute>
      <base-channel-power>550</base-channel-power>
      <admin-state>up</admin-state>
      <down-channel>
        <channel-index>1</channel-index>
        <admin-state>up</admin-state>
        <power-adjust>-2</power-adjust>
        <frequency>555000000</frequency>
        <rf-mute>false</rf-mute>
        <qam-alias>FN1_VOD1</qam-alias>
        <errp-advertising>true</errp-advertising>
        <erm-managed>
          <input-map-group-name>Group1</input-map-group-name>
          <phy-lock-parameters>interleaver</phy-lock-parameters>
          <allocation-type>video-only</allocation-type>
          <encryption-capability>
            <encryption-capability-index>1</encryption-capability-index>
            <ca-encryptor>motorola</ca-encryptor>
            <encryption-scheme>aes</encryption-scheme>
            <key-length>56</key-length>
          </encryption-capability>
          <erm-name>ERM-1</erm-name>
        </erm-managed>
        <video>
          <video-output-tsid>1</video-output-tsid>
          <video-phy-profile-index>1</video-phy-profile-index>
        </video>
      </down-channel>
      <down-channel>
        <channel-index>2</channel-index>
```

```
    <admin-state>up</admin-state>
    <power-adjust>-2</power-adjust>
    <frequency>561000000</frequency>
    <rf-mute>false</rf-mute>
    <qam-alias>FN1_VOD2</qam-alias>
    <errp-advertising>true</errp-advertising>
    <erm-managed>
      <input-map-group-name>Group2</input-map-group-name>
      <phy-lock-parameters>interleaver</phy-lock-parameters>
      <allocation-type>video-only</allocation-type>
      <encryption-capability>
        <encryption-capability-index>1</encryption-capability-index>
        <ca-encryptor>motorola</ca-encryptor>
        <encryption-scheme>aes</encryption-scheme>
        <key-length>56</key-length>
      </encryption-capability>
      <erm-name>ERM-1</erm-name>
    </erm-managed>
    <video>
      <video-output-tsid>2</video-output-tsid>
    </video>
  </down-channel>
  <down-channel>
    <channel-index>16</channel-index>
    <admin-state>up</admin-state>
    <power-adjust>0</power-adjust>
    <frequency>645000000</frequency>
    <rf-mute>false</rf-mute>
    <qam-alias>FN1_VOD16</qam-alias>
    <errp-advertising>true</errp-advertising>
    <erm-managed>
      <input-map-group-name>Group3</input-map-group-name>
      <phy-lock-parameters>interleaver</phy-lock-parameters>
      <allocation-type>video-only</allocation-type>
      <encryption-capability>
        <encryption-capability-index>1</encryption-capability-index>
        <ca-encryptor>motorola</ca-encryptor>
        <encryption-scheme>aes</encryption-scheme>
        <key-length>56</key-length>
      </encryption-capability>
      <erm-name>ERM-1</erm-name>
    </erm-managed>
    <video>
      <video-output-tsid>16</video-output-tsid>
    </video>
  </down-channel>
  <down-channel>
    <channel-index>17</channel-index>
    <admin-state>up</admin-state>
    <power-adjust>0</power-adjust>
    <frequency>651000000</frequency>
    <rf-mute>false</rf-mute>
    <qam-alias/>
    <errp-advertising>false</errp-advertising>
    <docsis>
      <id>0</id>
      <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
    </docsis>
  </down-channel>
  <down-channel>
    <channel-index>18</channel-index>
    <admin-state>up</admin-state>
    <power-adjust>0</power-adjust>
    <frequency>657000000</frequency>
    <rf-mute>false</rf-mute>
    <qam-alias/>
    <errp-advertising>false</errp-advertising>
    <docsis>
      <id>0</id>
      <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
      <docsis-phy-profile-index>1</docsis-phy-profile-index>
    </docsis>
  </down-channel>
  <down-channel>
    <channel-index>32</channel-index>
    <admin-state>up</admin-state>
    <power-adjust>2</power-adjust>
    <frequency>741000000</frequency>
```

```
    <rf-mute>false</rf-mute>
    <qam-alias/>
    <errp-advertising>false</errp-advertising>
    <docsis>
      <id>0</id>
      <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
    </docsis>
  </down-channel>
</ds-rf-port>
<ds-rf-port>
  <port-number>1</port-number>
  <rf-mute>false</rf-mute>
  <base-channel-power>550</base-channel-power>
  <admin-state>up</admin-state>
  <down-channel>
    <channel-index>1</channel-index>
    <admin-state>up</admin-state>
    <power-adjust>-2</power-adjust>
    <frequency>555000000</frequency>
    <rf-mute>false</rf-mute>
    <qam-alias>FN2_VOD1</qam-alias>
    <errp-advertising>true</errp-advertising>
    <erm-managed>
      <input-map-group-name>Group1</input-map-group-name>
      <phy-lock-parameters>interleaver</phy-lock-parameters>
      <allocation-type>video-only</allocation-type>
      <encryption-capability>
        <encryption-capability-index>1</encryption-capability-index>
        <ca-encryptor>motorola</ca-encryptor>
        <encryption-scheme>aes</encryption-scheme>
        <key-length>56</key-length>
      </encryption-capability>
      <erm-name>ERM-2</erm-name>
    </erm-managed>
    <video>
      <video-output-tsid>65</video-output-tsid>
    </video>
  </down-channel>
  <down-channel>
    <channel-index>2</channel-index>
    <admin-state>up</admin-state>
    <power-adjust>-2</power-adjust>
    <frequency>561000000</frequency>
    <rf-mute>false</rf-mute>
    <qam-alias>FN2_VOD2</qam-alias>
    <errp-advertising>true</errp-advertising>
    <erm-managed>
      <input-map-group-name>Group2</input-map-group-name>
      <phy-lock-parameters>interleaver</phy-lock-parameters>
      <allocation-type>video-only</allocation-type>
      <encryption-capability>
        <encryption-capability-index>1</encryption-capability-index>
        <ca-encryptor>motorola</ca-encryptor>
        <encryption-scheme>aes</encryption-scheme>
        <key-length>56</key-length>
      </encryption-capability>
      <erm-name>ERM-2</erm-name>
    </erm-managed>
    <video>
      <video-output-tsid>66</video-output-tsid>
    </video>
  </down-channel>
  <down-channel>
    <channel-index>16</channel-index>
    <admin-state>up</admin-state>
    <power-adjust>0</power-adjust>
    <frequency>645000000</frequency>
    <rf-mute>false</rf-mute>
    <qam-alias>FN2_VOD16</qam-alias>
    <errp-advertising>true</errp-advertising>
    <erm-managed>
      <input-map-group-name>Group3</input-map-group-name>
      <phy-lock-parameters>interleaver</phy-lock-parameters>
      <allocation-type>video-only</allocation-type>
      <encryption-capability>
        <encryption-capability-index>1</encryption-capability-index>
        <ca-encryptor>motorola</ca-encryptor>
        <encryption-scheme>aes</encryption-scheme>
```

```
        <key-length>56</key-length>
      </encryption-capability>
      <erm-name>ERM-2</erm-name>
    </erm-managed>
    <video>
      <video-output-tsid>80</video-output-tsid>
    </video>
  </down-channel>
  <down-channel>
    <channel-index>17</channel-index>
    <admin-state>up</admin-state>
    <power-adjust>0</power-adjust>
    <frequency>651000000</frequency>
    <rf-mute>false</rf-mute>
    <qam-alias/>
    <errp-advertising>false</errp-advertising>
    <docsis>
      <id>0</id>
      <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
    </docsis>
  </down-channel>
  <down-channel>
    <channel-index>18</channel-index>
    <admin-state>up</admin-state>
    <power-adjust>0</power-adjust>
    <frequency>657000000</frequency>
    <rf-mute>false</rf-mute>
    <qam-alias/>
    <errp-advertising>false</errp-advertising>
     <docsis>
      <id>0</id>
      <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
    </docsis>
  </down-channel>
  <down-channel>
    <channel-index>32</channel-index>
    <admin-state>up</admin-state>
    <power-adjust>2</power-adjust>
    <frequency>741000000</frequency>
    <rf-mute>false</rf-mute>
    <qam-alias/>
    <errp-advertising>false</errp-advertising>
    <docsis>
      <id>0</id>
      <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
    </docsis>
  </down-channel>
</ds-rf-port>
<ds-rf-port>
  <port-number>7</port-number>
  <rf-mute>false</rf-mute>
  <base-channel-power>550</base-channel-power>
  <admin-state>up</admin-state>
  <down-channel>
    <channel-index>1</channel-index>
    <admin-state>up</admin-state>
    <power-adjust>-2</power-adjust>
    <frequency>555000000</frequency>
    <rf-mute>false</rf-mute>
    <qam-alias>FN8_VOD1</qam-alias>
    <errp-advertising>true</errp-advertising>
    <erm-managed>
      <input-map-group-name>Group1</input-map-group-name>
      <phy-lock-parameters>interleaver</phy-lock-parameters>
      <allocation-type>video-only</allocation-type>
      <encryption-capability>
        <encryption-capability-index>1</encryption-capability-index>
        <ca-encryptor>motorola</ca-encryptor>
        <encryption-scheme>aes</encryption-scheme>
        <key-length>56</key-length>
      </encryption-capability>
      <erm-name>ERM-1</erm-name>
    </erm-managed>
    <video>
      <video-output-tsid>449</video-output-tsid>
    </video>
  </down-channel>
  <down-channel>
```

```
  <channel-index>2</channel-index>
  <admin-state>up</admin-state>
  <power-adjust>-2</power-adjust>
  <frequency>561000000</frequency>
  <rf-mute>false</rf-mute>
  <qam-alias>FN8_VOD2</qam-alias>
  <errp-advertising>true</errp-advertising>
  <erm-managed>
    <input-map-group-name>Group2</input-map-group-name>
    <phy-lock-parameters>interleaver</phy-lock-parameters>
    <allocation-type>video-only</allocation-type>
    <encryption-capability>
      <encryption-capability-index>1</encryption-capability-index>
      <ca-encryptor>motorola</ca-encryptor>
      <encryption-scheme>aes</encryption-scheme>
      <key-length>56</key-length>
    </encryption-capability>
    <erm-name>ERM-1</erm-name>
  </erm-managed>
  <video>
    <video-output-tsid>450</video-output-tsid>
  </video>
</down-channel>
<down-channel>
  <channel-index>16</channel-index>
  <admin-state>up</admin-state>
  <power-adjust>0</power-adjust>
  <frequency>645000000</frequency>
  <rf-mute>false</rf-mute>
  <qam-alias>FN8_VOD16</qam-alias>
  <errp-advertising>true</errp-advertising>
  <erm-managed>
    <input-map-group-name>Group3</input-map-group-name>
    <phy-lock-parameters>interleaver</phy-lock-parameters>
    <allocation-type>video-only</allocation-type>
    <encryption-capability>
      <encryption-capability-index>1</encryption-capability-index>
      <ca-encryptor>motorola</ca-encryptor>
      <encryption-scheme>aes</encryption-scheme>
      <key-length>56</key-length>
    </encryption-capability>
    <erm-name>ERM-1</erm-name>
  </erm-managed>
  <video>
    <video-output-tsid>464</video-output-tsid>
  </video>
</down-channel>
<down-channel>
  <channel-index>17</channel-index>
  <admin-state>up</admin-state>
  <power-adjust>0</power-adjust>
  <frequency>651000000</frequency>
  <rf-mute>false</rf-mute>
  <qam-alias/>
  <errp-advertising>false</errp-advertising>
  <docsis>
    <id>0</id>
    <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
  </docsis>
</down-channel>
<down-channel>
  <channel-index>18</channel-index>
  <admin-state>up</admin-state>
  <power-adjust>0</power-adjust>
  <frequency>657000000</frequency>
  <rf-mute>false</rf-mute>
  <qam-alias/>
  <errp-advertising>false</errp-advertising>
  <docsis>
    <id>0</id>
    <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
  </docsis>
</down-channel>
<down-channel>
  <channel-index>32</channel-index>
  <admin-state>up</admin-state>
  <power-adjust>2</power-adjust>
  <frequency>741000000</frequency>
```

```
        <rf-mute>false</rf-mute>
        <qam-alias/>
        <errp-advertising>false</errp-advertising>
        <docsis>
          <id>0</id>
          <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
        </docsis>
      </down-channel>
    </ds-rf-port>
   </rf-line-card>
</slot>
<slot>
  <slot-number>2</slot-number>
   <rf-line-card>
    <rf-card>
     <line-card-name>Downstream RF Line Card Spare</line-card-name>
     <admin-state>up</admin-state>
    </rf-card>
   </rf-line-card>
</slot>
<slot>
  <slot-number>3</slot-number>
  <rf-line-card>
    <rf-card>
      <line-card-name>Downstream RF Line Card 3</line-card-name>
      <admin-state>up</admin-state>
      <protected-by>2</protected-by>
    </rf-card>
    <encryptor>
      <encryptor-index>1</encryptor-index>
      <ca-encryptor-type>motorola</ca-encryptor-type>
      <ecm-timeout>10</ecm-timeout>
      <clear-stream-timeout>10</clear-stream-timeout>
      <ecmg-usage>
        <ecmg-usage-index>1</ecmg-usage-index>
        <priority>1</priority>
        <ecmg-ref>1</ecmg-ref>
      </ecmg-usage>
    </encryptor>
    <ds-rf-port>
      <port-number>0</port-number>
      <rf-mute>false</rf-mute>
      <base-channel-power>550</base-channel-power>
      <admin-state>up</admin-state>
      <down-channel>
        <channel-index>1</channel-index>
        <admin-state>up</admin-state>
        <power-adjust>-2</power-adjust>
        <frequency>555000000</frequency>
        <rf-mute>false</rf-mute>
        <qam-alias>FN9_VOD1</qam-alias>
        <errp-advertising>true</errp-advertising>
        <erm-managed>
          <input-map-group-name>Group1</input-map-group-name>
          <phy-lock-parameters>interleaver</phy-lock-parameters>
          <allocation-type>video-only</allocation-type>
          <encryption-capability>
            <encryption-capability-index>1</encryption-capability-index>
            <ca-encryptor>motorola</ca-encryptor>
            <encryption-scheme>aes</encryption-scheme>
            <key-length>56</key-length>
          </encryption-capability>
          <erm-name>ERM-1</erm-name>
        </erm-managed>
        <video>
          <video-output-tsid>513</video-output-tsid>
        </video>
      </down-channel>
      <down-channel>
        <channel-index>2</channel-index>
        <admin-state>up</admin-state>
        <power-adjust>-2</power-adjust>
        <frequency>561000000</frequency>
        <rf-mute>false</rf-mute>
        <qam-alias>FN9_VOD2</qam-alias>
        <errp-advertising>true</errp-advertising>
        <erm-managed>
          <input-map-group-name>Group2</input-map-group-name>
```

```
      <phy-lock-parameters>interleaver</phy-lock-parameters>
      <allocation-type>video-only</allocation-type>
      <encryption-capability>
        <encryption-capability-index>1</encryption-capability-index>
        <ca-encryptor>motorola</ca-encryptor>
        <encryption-scheme>aes</encryption-scheme>
        <key-length>56</key-length>
      </encryption-capability>
      <erm-name>ERM-1</erm-name>
    </erm-managed>
    <video>
      <video-output-tsid>514</video-output-tsid>
    </video>
  </down-channel>
  <down-channel>
    <channel-index>16</channel-index>
    <admin-state>up</admin-state>
    <power-adjust>0</power-adjust>
    <frequency>645000000</frequency>
    <rf-mute>false</rf-mute>
    <qam-alias>FN9_VOD16</qam-alias>
    <errp-advertising>true</errp-advertising>
    <erm-managed>
      <input-map-group-name>Group3</input-map-group-name>
      <phy-lock-parameters>interleaver</phy-lock-parameters>
      <allocation-type>video-only</allocation-type>
      <encryption-capability>
        <encryption-capability-index>1</encryption-capability-index>
        <ca-encryptor>motorola</ca-encryptor>
        <encryption-scheme>aes</encryption-scheme>
        <key-length>56</key-length>
      </encryption-capability>
      <erm-name>ERM-1</erm-name>
    </erm-managed>
    <video>
      <video-output-tsid>528</video-output-tsid>
    </video>
  </down-channel>
  <down-channel>
    <channel-index>17</channel-index>
    <admin-state>up</admin-state>
    <power-adjust>0</power-adjust>
    <frequency>651000000</frequency>
    <rf-mute>false</rf-mute>
    <qam-alias/>
    <errp-advertising>false</errp-advertising>
    <docsis>
      <id>0</id>
      <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
    </docsis>
  </down-channel>
  <down-channel>
    <channel-index>18</channel-index>
    <admin-state>up</admin-state>
    <power-adjust>0</power-adjust>
    <frequency>657000000</frequency>
    <rf-mute>false</rf-mute>
    <qam-alias/>
    <errp-advertising>false</errp-advertising>
    <docsis>
      <id>0</id>
      <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
    </docsis>
  </down-channel>
  <down-channel>
    <channel-index>32</channel-index>
    <admin-state>up</admin-state>
    <power-adjust>2</power-adjust>
    <frequency>741000000</frequency>
    <rf-mute>false</rf-mute>
    <qam-alias/>
    <errp-advertising>false</errp-advertising>
     <docsis>
      <id>0</id>
      <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
    </docsis>
  </down-channel>
</ds-rf-port>
```

```
<ds-rf-port>
  <port-number>1</port-number>
  <rf-mute>false</rf-mute>
  <base-channel-power>550</base-channel-power>
  <admin-state>up</admin-state>
  <down-channel>
    <channel-index>1</channel-index>
    <admin-state>up</admin-state>
    <power-adjust>-2</power-adjust>
    <frequency>555000000</frequency>
    <rf-mute>false</rf-mute>
    <qam-alias>FN10_VOD1</qam-alias>
    <errp-advertising>true</errp-advertising>
    <erm-managed>
      <input-map-group-name>Group1</input-map-group-name>
      <phy-lock-parameters>interleaver</phy-lock-parameters>
      <allocation-type>video-only</allocation-type>
      <encryption-capability>
        <encryption-capability-index>1</encryption-capability-index>
        <ca-encryptor>motorola</ca-encryptor>
        <encryption-scheme>aes</encryption-scheme>
        <key-length>56</key-length>
      </encryption-capability>
      <erm-name>ERM-1</erm-name>
    </erm-managed>
    <video>
      <video-output-tsid>577</video-output-tsid>
    </video>
  </down-channel>
  <down-channel>
    <channel-index>2</channel-index>
    <admin-state>up</admin-state>
    <power-adjust>-2</power-adjust>
    <frequency>561000000</frequency>
    <rf-mute>false</rf-mute>
    <qam-alias>FN10_VOD2</qam-alias>
    <errp-advertising>true</errp-advertising>
    <erm-managed>
      <input-map-group-name>Group2</input-map-group-name>
      <phy-lock-parameters>interleaver</phy-lock-parameters>
      <allocation-type>video-only</allocation-type>
      <encryption-capability>
        <encryption-capability-index>1</encryption-capability-index>
        <ca-encryptor>motorola</ca-encryptor>
        <encryption-scheme>aes</encryption-scheme>
        <key-length>56</key-length>
      </encryption-capability>
      <erm-name>ERM-1</erm-name>
    </erm-managed>
    <video>
      <video-output-tsid>578</video-output-tsid>
    </video>
  </down-channel>
  <down-channel>
    <channel-index>16</channel-index>
    <admin-state>up</admin-state>
    <power-adjust>0</power-adjust>
    <frequency>645000000</frequency>
    <rf-mute>false</rf-mute>
    <qam-alias>FN10_VOD16</qam-alias>
    <errp-advertising>true</errp-advertising>
    <erm-managed>
      <input-map-group-name>Group3</input-map-group-name>
      <phy-lock-parameters>interleaver</phy-lock-parameters>
      <allocation-type>video-only</allocation-type>
      <encryption-capability>
        <encryption-capability-index>1</encryption-capability-index>
        <ca-encryptor>motorola</ca-encryptor>
        <encryption-scheme>aes</encryption-scheme>
        <key-length>56</key-length>
      </encryption-capability>
      <erm-name>ERM-1</erm-name>
    </erm-managed>
    <video>
      <video-output-tsid>598</video-output-tsid>
    </video>
  </down-channel>
  <down-channel>
```

```
      <channel-index>17</channel-index>
      <admin-state>up</admin-state>
      <power-adjust>0</power-adjust>
      <frequency>651000000</frequency>
      <rf-mute>false</rf-mute>
      <qam-alias/>
      <errp-advertising>false</errp-advertising>
      <docsis>
        <id>0</id>
        <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
      </docsis>
    </down-channel>
    <down-channel>
      <channel-index>18</channel-index>
      <admin-state>up</admin-state>
      <power-adjust>0</power-adjust>
      <frequency>657000000</frequency>
      <rf-mute>false</rf-mute>
      <qam-alias/>
      <errp-advertising>false</errp-advertising>
      <docsis>
        <id>0</id>
        <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
      </docsis>
    </down-channel>
    <down-channel>
      <channel-index>32</channel-index>
      <admin-state>up</admin-state>
      <power-adjust>2</power-adjust>
      <frequency>741000000</frequency>
      <rf-mute>false</rf-mute>
      <qam-alias/>
      <errp-advertising>false</errp-advertising>
      <docsis>
        <id>0</id>
        <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
      </docsis>
    </down-channel>
  </ds-rf-port>
  <ds-rf-port>
    <port-number>7</port-number>
    <rf-mute>false</rf-mute>
    <base-channel-power>550</base-channel-power>
    <admin-state>up</admin-state>
    <down-channel>
      <channel-index>1</channel-index>
      <admin-state>up</admin-state>
      <power-adjust>-2</power-adjust>
      <frequency>555000000</frequency>
      <rf-mute>false</rf-mute>
      <qam-alias>FN16_VOD1</qam-alias>
      <errp-advertising>true</errp-advertising>
      <erm-managed>
        <input-map-group-name>Group1</input-map-group-name>
        <phy-lock-parameters>interleaver</phy-lock-parameters>
        <allocation-type>video-only</allocation-type>
        <encryption-capability>
          <encryption-capability-index>1</encryption-capability-index>
          <ca-encryptor>motorola</ca-encryptor>
          <encryption-scheme>aes</encryption-scheme>
          <key-length>56</key-length>
        </encryption-capability>
        <erm-name>ERM-1</erm-name>
      </erm-managed>
      <video>
        <video-output-tsid>961</video-output-tsid>
      </video>
    </down-channel>
    <down-channel>
      <channel-index>2</channel-index>
      <admin-state>up</admin-state>
      <power-adjust>-2</power-adjust>
      <frequency>561000000</frequency>
      <rf-mute>false</rf-mute>
      <qam-alias>FN16_VOD2</qam-alias>
      <errp-advertising>true</errp-advertising>
      <erm-managed>
        <input-map-group-name>Group2</input-map-group-name>
```

```
      <phy-lock-parameters>interleaver</phy-lock-parameters>
      <allocation-type>video-only</allocation-type>
      <encryption-capability>
        <encryption-capability-index>1</encryption-capability-index>
        <ca-encryptor>motorola</ca-encryptor>
        <encryption-scheme>aes</encryption-scheme>
        <key-length>56</key-length>
      </encryption-capability>
      <erm-name>ERM-1</erm-name>
    </erm-managed>
    <video>
      <video-output-tsid>962</video-output-tsid>
    </video>
  </down-channel>
  <down-channel>
    <channel-index>16</channel-index>
    <admin-state>up</admin-state>
    <power-adjust>0</power-adjust>
    <frequency>645000000</frequency>
    <rf-mute>false</rf-mute>
    <qam-alias>FN16_VOD16</qam-alias>
    <errp-advertising>true</errp-advertising>
    <erm-managed>
      <input-map-group-name>Group3</input-map-group-name>
      <phy-lock-parameters>interleaver</phy-lock-parameters>
      <allocation-type>video-only</allocation-type>
      <encryption-capability>
        <encryption-capability-index>1</encryption-capability-index>
        <ca-encryptor>motorola</ca-encryptor>
        <encryption-scheme>aes</encryption-scheme>
        <key-length>56</key-length>
      </encryption-capability>
      <erm-name>ERM-1</erm-name>
    </erm-managed>
    <video>
      <video-output-tsid>976</video-output-tsid>
    </video>
  </down-channel>
  <down-channel>
    <channel-index>17</channel-index>
    <admin-state>up</admin-state>
    <power-adjust>0</power-adjust>
    <frequency>651000000</frequency>
    <rf-mute>false</rf-mute>
    <qam-alias/>
    <errp-advertising>false</errp-advertising>
    <docsis>
      <id>0</id>
      <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
    </docsis>
  </down-channel>
  <down-channel>
    <channel-index>18</channel-index>
    <admin-state>up</admin-state>
    <power-adjust>0</power-adjust>
    <frequency>657000000</frequency>
    <rf-mute>false</rf-mute>
    <qam-alias/>
    <errp-advertising>false</errp-advertising>
    <docsis>
      <id>0</id>
      <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
    </docsis>
  </down-channel>
  <down-channel>
    <channel-index>32</channel-index>
    <admin-state>up</admin-state>
    <power-adjust>2</power-adjust>
    <frequency>741000000</frequency>
    <rf-mute>false</rf-mute>
    <qam-alias/>
    <errp-advertising>false</errp-advertising>
    <docsis>
      <id>0</id>
      <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
    </docsis>
  </down-channel>
</ds-rf-port>
```

```
  </rf-line-card>
</slot>
<slot>
  <slot-number>6</slot-number>
  <sre-line-card>
    <sre-card>
      <line-card-name>SRE 6</line-card-name>
      <admin-state>up</admin-state>
      <protected-by>7</protected-by>
    </sre-card>
    <one-gb-ethernet-port>
      <port-number>0</port-number>
      <admin-state>up</admin-state>
      <ip-interface>
        <ip-interface-name>eth6/0</ip-interface-name>
        <primary-ipv4>
          <ip-address>10.10.10.99/32</ip-address>
        </primary-ipv4>
        <ingress-acl>acl1</ingress-acl>
      </ip-interface>
      <speed>auto</speed>
    </one-gb-ethernet-port>
    <ten-gb-ethernet-port>
      <port-number>3</port-number>
      <admin-state>up</admin-state>
      <ip-interface>
        <ip-interface-name>eth6/3</ip-interface-name>
        <primary-ipv4>
          <ip-address>66.77.88.99/32</ip-address>
        </primary-ipv4>
        <egress-acl>acl2</egress-acl>
      </ip-interface>
    </ten-gb-ethernet-port>
  </sre-line-card>
</slot>
<slot>
  <slot-number>7</slot-number>
   <sre-line-card>
    <sre-card>
      <line-card-name>SRE 7</line-card-name>
      <admin-state>up</admin-state>
      <protected-by>6</protected-by>
    </sre-card>
    <one-gb-ethernet-port>
      <port-number>0</port-number>
      <admin-state>up</admin-state>
      <up-down-trap-enabled>true</up-down-trap-enabled>
      <ip-interface>
        <ip-interface-name>eth7/0</ip-interface-name>
        <primary-ipv4>
          <ip-address>10.10.10.100/32</ip-address>
        </primary-ipv4>
      </ip-interface>
      <speed>auto</speed>
    </one-gb-ethernet-port>
    <ten-gb-ethernet-port>
      <port-number>6</port-number>
      <admin-state>up</admin-state>
      <up-down-trap-enabled>true</up-down-trap-enabled>
      <ip-interface>
        <ip-interface-name>eth7/6</ip-interface-name>
        <primary-ipv4>
          <ip-address>66.77.88.100/32</ip-address>
        </primary-ipv4>
      </ip-interface>
    </ten-gb-ethernet-port>
  </sre-line-card>
</slot>
<slot>
  <slot-number>8</slot-number>
  <epon-line-card>
    <epon-card>
      <line-card-name>PON 8</line-card-name>
      <admin-state>up</admin-state>
    </epon-card>
  </epon-line-card>
</slot>
<slot>
```

```
<slot-number>9</slot-number>
<rf-line-card>
  <rf-card>
    <line-card-name>Upstream RF Line Card 9</line-card-name>
    <admin-state>up</admin-state>
    <protected-by>10</protected-by>
  </rf-card>
  <us-rf-port>
    <port-number>0</port-number>
    <admin-state>up</admin-state>
    <upstream-physical-channel>
      <channel-index>0</channel-index>
      <admin-state>up</admin-state>
      <frequency>6800000</frequency>
      <width>3200000</width>
      <power-level>0</power-level>
      <upstream-logical-channel>
        <upstream-logical-channel-index>0</upstream-logical-channel-index>
        <admin-state>up</admin-state>
        <channel-id>0</channel-id>
        <slot-size>2</slot-size>
        <ranging-backoff-start>2</ranging-backoff-start>
        <ranging-backoff-end>8</ranging-backoff-end>
        <transmit-backoff-start>2</transmit-backoff-start>
        <transmit-backoff-end>8</transmit-backoff-end>
        <pre-equalization-enable>true</pre-equalization-enable>
        <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
        <power-level-adjust>0</power-level-adjust>
        <modulation>1</modulation>
        <atdma-logical-channel/>
      </upstream-logical-channel>
    </upstream-physical-channel>
    <upstream-physical-channel>
      <channel-index>1</channel-index>
      <admin-state>up</admin-state>
      <frequency>11200000</frequency>
      <width>6400000</width>
      <power-level>0</power-level>
      <upstream-logical-channel>
        <upstream-logical-channel-index>0</upstream-logical-channel-index>
        <admin-state>up</admin-state>
        <channel-id>0</channel-id>
        <slot-size>2</slot-size>
        <ranging-backoff-start>2</ranging-backoff-start>
        <ranging-backoff-end>8</ranging-backoff-end>
        <transmit-backoff-start>2</transmit-backoff-start>
        <transmit-backoff-end>8</transmit-backoff-end>
        <pre-equalization-enable>true</pre-equalization-enable>
        <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
        <power-level-adjust>0</power-level-adjust>
        <modulation>1</modulation>
        <atdma-logical-channel/>
      </upstream-logical-channel>
    </upstream-physical-channel>
    <upstream-physical-channel>
      <channel-index>5</channel-index>
      <admin-state>up</admin-state>
      <frequency>36800000</frequency>
      <width>6400000</width>
      <power-level>0</power-level>
      <upstream-logical-channel>
        <upstream-logical-channel-index>0</upstream-logical-channel-index>
        <admin-state>up</admin-state>
        <channel-id>0</channel-id>
        <slot-size>2</slot-size>
        <ranging-backoff-start>2</ranging-backoff-start>
        <ranging-backoff-end>8</ranging-backoff-end>
        <transmit-backoff-start>2</transmit-backoff-start>
        <transmit-backoff-end>8</transmit-backoff-end>
        <pre-equalization-enable>true</pre-equalization-enable>
        <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
        <power-level-adjust>0</power-level-adjust>
        <modulation>1</modulation>
        <atdma-logical-channel/>
     </upstream-logical-channel>
    </upstream-physical-channel>
  </us-rf-port>
  <us-rf-port>
```

```
  <port-number>1</port-number>
  <admin-state>up</admin-state>
  <upstream-physical-channel>
    <channel-index>0</channel-index>
    <admin-state>up</admin-state>
    <frequency>6800000</frequency>
    <width>3200000</width>
    <power-level>0</power-level>
    <upstream-logical-channel>
      <upstream-logical-channel-index>0</upstream-logical-channel-index>
      <admin-state>up</admin-state>
      <channel-id>0</channel-id>
      <slot-size>2</slot-size>
      <ranging-backoff-start>2</ranging-backoff-start>
      <ranging-backoff-end>8</ranging-backoff-end>
      <transmit-backoff-start>2</transmit-backoff-start>
      <transmit-backoff-end>8</transmit-backoff-end>
      <pre-equalization-enable>true</pre-equalization-enable>
      <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
      <power-level-adjust>0</power-level-adjust>
      <modulation>1</modulation>
      <atdma-logical-channel/>
    </upstream-logical-channel>
  </upstream-physical-channel>
  <upstream-physical-channel>
    <channel-index>1</channel-index>
    <admin-state>up</admin-state>
    <frequency>11200000</frequency>
    <width>6400000</width>
    <power-level>0</power-level>
    <upstream-logical-channel>
      <upstream-logical-channel-index>0</upstream-logical-channel-index>
      <admin-state>up</admin-state>
      <channel-id>0</channel-id>
      <slot-size>2</slot-size>
      <ranging-backoff-start>2</ranging-backoff-start>
      <ranging-backoff-end>8</ranging-backoff-end>
      <transmit-backoff-start>2</transmit-backoff-start>
      <transmit-backoff-end>8</transmit-backoff-end>
      <pre-equalization-enable>true</pre-equalization-enable>
      <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
      <power-level-adjust>0</power-level-adjust>
      <modulation>1</modulation>
      <atdma-logical-channel/>
    </upstream-logical-channel>
  </upstream-physical-channel>
  <upstream-physical-channel>
    <channel-index>5</channel-index>
    <admin-state>up</admin-state>
    <frequency>36800000</frequency>
    <width>6400000</width>
    <power-level>0</power-level>
    <upstream-logical-channel>
      <upstream-logical-channel-index>0</upstream-logical-channel-index>
      <admin-state>up</admin-state>
      <channel-id>0</channel-id>
      <slot-size>2</slot-size>
      <ranging-backoff-start>2</ranging-backoff-start>
      <ranging-backoff-end>8</ranging-backoff-end>
      <transmit-backoff-start>2</transmit-backoff-start>
      <transmit-backoff-end>8</transmit-backoff-end>
      <pre-equalization-enable>true</pre-equalization-enable>
      <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
      <power-level-adjust>0</power-level-adjust>
      <modulation>1</modulation>
      <atdma-logical-channel/>
    </upstream-logical-channel>
  </upstream-physical-channel>
</us-rf-port>
<us-rf-port>
  <port-number>7</port-number>
  <admin-state>up</admin-state>
  <upstream-physical-channel>
    <channel-index>0</channel-index>
    <admin-state>up</admin-state>
    <frequency>6800000</frequency>
    <width>3200000</width>
    <power-level>0</power-level>
```

```
        <upstream-logical-channel>
          <upstream-logical-channel-index>0</upstream-logical-channel-index>
          <admin-state>up</admin-state>
          <channel-id>0</channel-id>
          <slot-size>2</slot-size>
          <ranging-backoff-start>2</ranging-backoff-start>
          <ranging-backoff-end>8</ranging-backoff-end>
          <transmit-backoff-start>2</transmit-backoff-start>
          <transmit-backoff-end>8</transmit-backoff-end>
          <pre-equalization-enable>true</pre-equalization-enable>
          <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
          <power-level-adjust>0</power-level-adjust>
          <modulation>1</modulation>
          <atdma-logical-channel/>
        </upstream-logical-channel>
      </upstream-physical-channel>
      <upstream-physical-channel>
        <channel-index>1</channel-index>
        <admin-state>up</admin-state>
        <frequency>11200000</frequency>
        <width>6400000</width>
        <power-level>0</power-level>
        <upstream-logical-channel>
          <upstream-logical-channel-index>0</upstream-logical-channel-index>
          <admin-state>up</admin-state>
          <channel-id>0</channel-id>
          <slot-size>2</slot-size>
          <ranging-backoff-start>2</ranging-backoff-start>
          <ranging-backoff-end>8</ranging-backoff-end>
          <transmit-backoff-start>2</transmit-backoff-start>
          <transmit-backoff-end>8</transmit-backoff-end>
          <pre-equalization-enable>true</pre-equalization-enable>
          <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
          <power-level-adjust>0</power-level-adjust>
          <modulation>1</modulation>
          <atdma-logical-channel/>
        </upstream-logical-channel>
      </upstream-physical-channel>
      <upstream-physical-channel>
        <channel-index>5</channel-index>
        <admin-state>up</admin-state>
        <frequency>36800000</frequency>
        <width>6400000</width>
        <power-level>0</power-level>
        <upstream-logical-channel>
          <upstream-logical-channel-index>0</upstream-logical-channel-index>
          <admin-state>up</admin-state>
          <channel-id>0</channel-id>
          <slot-size>2</slot-size>
          <ranging-backoff-start>2</ranging-backoff-start>
          <ranging-backoff-end>8</ranging-backoff-end>
          <transmit-backoff-start>2</transmit-backoff-start>
          <transmit-backoff-end>8</transmit-backoff-end>
          <pre-equalization-enable>true</pre-equalization-enable>
          <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
          <power-level-adjust>0</power-level-adjust>
          <modulation>1</modulation>
          <atdma-logical-channel/>
        </upstream-logical-channel>
      </upstream-physical-channel>
    </us-rf-port>
  </rf-line-card>
</slot>
<slot>
  <slot-number>10</slot-number>
   <rf-line-card>
    <rf-card>
      <line-card-name>Upstream RF Line Card Spare</line-card-name>
      <admin-state>up</admin-state>
    </rf-card>
   </rf-line-card>
</slot>
<slot>
  <slot-number>11</slot-number>
   <rf-line-card>
    <rf-card>
      <line-card-name>Upstream RF Line Card 11</line-card-name>
      <admin-state>up</admin-state>
```

```
  <protected-by>10</protected-by>
</rf-card>
<us-rf-port>
  <port-number>0</port-number>
  <admin-state>up</admin-state>
  <upstream-physical-channel>
    <channel-index>0</channel-index>
    <admin-state>up</admin-state>
    <frequency>6800000</frequency>
    <width>3200000</width>
    <power-level>0</power-level>
    <upstream-logical-channel>
      <upstream-logical-channel-index>0</upstream-logical-channel-index>
      <admin-state>up</admin-state>
      <channel-id>1</channel-id>
      <slot-size>2</slot-size>
      <ranging-backoff-start>12</ranging-backoff-start>
      <ranging-backoff-end>16</ranging-backoff-end>
      <transmit-backoff-start>12</transmit-backoff-start>
      <transmit-backoff-end>16</transmit-backoff-end>
      <pre-equalization-enable>true</pre-equalization-enable>
      <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
      <power-level-adjust>0</power-level-adjust>
      <modulation>1</modulation>
      <atdma-logical-channel/>
    </upstream-logical-channel>
  </upstream-physical-channel>
  <upstream-physical-channel>
    <channel-index>1</channel-index>
    <admin-state>up</admin-state>
    <frequency>11200000</frequency>
    <width>6400000</width>
    <power-level>0</power-level>
    <upstream-logical-channel>
      <upstream-logical-channel-index>0</upstream-logical-channel-index>
      <admin-state>up</admin-state>
      <channel-id>2</channel-id>
      <slot-size>4</slot-size>
      <ranging-backoff-start>2</ranging-backoff-start>
      <ranging-backoff-end>8</ranging-backoff-end>
      <transmit-backoff-start>5</transmit-backoff-start>
      <transmit-backoff-end>8</transmit-backoff-end>
      <pre-equalization-enable>true</pre-equalization-enable>
      <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
      <power-level-adjust>0</power-level-adjust>
      <modulation>1</modulation>
      <atdma-logical-channel/>
    </upstream-logical-channel>
  </upstream-physical-channel>
  <upstream-physical-channel>
    <channel-index>5</channel-index>
    <admin-state>up</admin-state>
    <frequency>36800000</frequency>
    <width>6400000</width>
    <power-level>0</power-level>
    <upstream-logical-channel>
      <upstream-logical-channel-index>0</upstream-logical-channel-index>
      <admin-state>up</admin-state>
      <channel-id>2</channel-id>
      <slot-size>4</slot-size>
      <ranging-backoff-start>2</ranging-backoff-start>
      <ranging-backoff-end>8</ranging-backoff-end>
      <transmit-backoff-start>5</transmit-backoff-start>
      <transmit-backoff-end>8</transmit-backoff-end>
      <pre-equalization-enable>true</pre-equalization-enable>
      <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
      <power-level-adjust>0</power-level-adjust>
      <modulation>1</modulation>
      <atdma-logical-channel/>
    </upstream-logical-channel>
  </upstream-physical-channel>
</us-rf-port>
<us-rf-port>
  <port-number>1</port-number>
  <admin-state>up</admin-state>
  <upstream-physical-channel>
    <channel-index>0</channel-index>
    <admin-state>up</admin-state>
```

```
      <frequency>6800000</frequency>
      <width>3200000</width>
      <power-level>0</power-level>
      <upstream-logical-channel>
        <upstream-logical-channel-index>0</upstream-logical-channel-index>
        <admin-state>up</admin-state>
        <channel-id>0</channel-id>
        <slot-size>2</slot-size>
        <ranging-backoff-start>2</ranging-backoff-start>
        <ranging-backoff-end>8</ranging-backoff-end>
        <transmit-backoff-start>2</transmit-backoff-start>
        <transmit-backoff-end>8</transmit-backoff-end>
        <pre-equalization-enable>true</pre-equalization-enable>
        <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
        <power-level-adjust>0</power-level-adjust>
        <modulation>1</modulation>
        <atdma-logical-channel/>
     </upstream-logical-channel>
    </upstream-physical-channel>
    <upstream-physical-channel>
      <channel-index>1</channel-index>
      <admin-state>up</admin-state>
      <frequency>11200000</frequency>
      <width>6400000</width>
      <power-level>0</power-level>
      <upstream-logical-channel>
        <upstream-logical-channel-index>0</upstream-logical-channel-index>
        <admin-state>up</admin-state>
        <channel-id>0</channel-id>
        <slot-size>2</slot-size>
        <ranging-backoff-start>2</ranging-backoff-start>
        <ranging-backoff-end>8</ranging-backoff-end>
        <transmit-backoff-start>2</transmit-backoff-start>
        <transmit-backoff-end>8</transmit-backoff-end>
        <pre-equalization-enable>true</pre-equalization-enable>
        <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
        <power-level-adjust>0</power-level-adjust>
        <modulation>1</modulation>
        <atdma-logical-channel/>
      </upstream-logical-channel>
    </upstream-physical-channel>
    <upstream-physical-channel>
      <channel-index>5</channel-index>
      <admin-state>up</admin-state>
      <frequency>36800000</frequency>
      <width>6400000</width>
      <power-level>0</power-level>
      <upstream-logical-channel>
        <upstream-logical-channel-index>0</upstream-logical-channel-index>
        <admin-state>up</admin-state>
        <channel-id>0</channel-id>
        <slot-size>2</slot-size>
        <ranging-backoff-start>2</ranging-backoff-start>
        <ranging-backoff-end>8</ranging-backoff-end>
        <transmit-backoff-start>2</transmit-backoff-start>
        <transmit-backoff-end>8</transmit-backoff-end>
        <pre-equalization-enable>true</pre-equalization-enable>
        <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
        <power-level-adjust>0</power-level-adjust>
        <modulation>2</modulation>
        <atdma-logical-channel/>
      </upstream-logical-channel>
    </upstream-physical-channel>
  </us-rf-port>
  <us-rf-port>
    <port-number>7</port-number>
    <admin-state>up</admin-state>
    <upstream-physical-channel>
      <channel-index>0</channel-index>
      <admin-state>up</admin-state>
      <frequency>6800000</frequency>
      <width>3200000</width>
      <power-level>0</power-level>
      <upstream-logical-channel>
        <upstream-logical-channel-index>0</upstream-logical-channel-index>
        <admin-state>up</admin-state>
        <channel-id>0</channel-id>
        <slot-size>2</slot-size>
```

```
          <ranging-backoff-start>2</ranging-backoff-start>
          <ranging-backoff-end>8</ranging-backoff-end>
          <transmit-backoff-start>2</transmit-backoff-start>
          <transmit-backoff-end>8</transmit-backoff-end>
          <pre-equalization-enable>true</pre-equalization-enable>
          <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
          <power-level-adjust>0</power-level-adjust>
          <modulation>1</modulation>
          <atdma-logical-channel/>
        </upstream-logical-channel>
      </upstream-physical-channel>
      <upstream-physical-channel>
        <channel-index>1</channel-index>
        <admin-state>up</admin-state>
        <frequency>11200000</frequency>
        <width>6400000</width>
        <power-level>0</power-level>
        <upstream-logical-channel>
          <upstream-logical-channel-index>0</upstream-logical-channel-index>
          <admin-state>up</admin-state>
          <channel-id>0</channel-id>
          <slot-size>2</slot-size>
          <ranging-backoff-start>2</ranging-backoff-start>
          <ranging-backoff-end>8</ranging-backoff-end>
          <transmit-backoff-start>2</transmit-backoff-start>
          <transmit-backoff-end>8</transmit-backoff-end>
          <pre-equalization-enable>true</pre-equalization-enable>
          <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
          <power-level-adjust>0</power-level-adjust>
          <modulation>1</modulation>
          <atdma-logical-channel/>
        </upstream-logical-channel>
      </upstream-physical-channel>
      <upstream-physical-channel>
        <channel-index>5</channel-index>
        <admin-state>up</admin-state>
        <frequency>36800000</frequency>
        <width>6400000</width>
        <power-level>0</power-level>
        <upstream-logical-channel>
          <upstream-logical-channel-index>0</upstream-logical-channel-index>
          <admin-state>up</admin-state>
          <channel-id>0</channel-id>
          <slot-size>2</slot-size>
          <ranging-backoff-start>2</ranging-backoff-start>
          <ranging-backoff-end>8</ranging-backoff-end>
          <transmit-backoff-start>2</transmit-backoff-start>
          <transmit-backoff-end>8</transmit-backoff-end>
          <pre-equalization-enable>true</pre-equalization-enable>
          <provisioned-attribute-mask>bonded</provisioned-attribute-mask>
          <power-level-adjust>0</power-level-adjust>
          <modulation>1</modulation>
          <atdma-logical-channel/>
        </upstream-logical-channel>
      </upstream-physical-channel>
    </us-rf-port>
   </rf-line-card>
</slot>
<video-phy-profile>
  <phy-index>0</phy-index>
  <modulation>qam256</modulation>
  <interleaver-depth>fecI128J1</interleaver-depth>
  <downstream-phy-standard>j83annexB</downstream-phy-standard>
  <spectrum-inversion>false</spectrum-inversion>
</video-phy-profile>
<video-phy-profile>
  <phy-index>1</phy-index>
  <modulation>qam64</modulation>
  <interleaver-depth>fecI128J4</interleaver-depth>
  <spectrum-inversion>true</spectrum-inversion>
  <symbol-rate-override>40000000</symbol-rate-override>
</video-phy-profile>
<docsis-phy-profile>
  <phy-index>0</phy-index>
  <modulation>qam256</modulation>
  <interleaver-depth>fecI32J4</interleaver-depth>
  <downstream-phy-standard>j83annexB</downstream-phy-standard>
</docsis-phy-profile>
```

```
  <docsis-phy-profile>
    <phy-index>1</phy-index>
    <modulation>qam64</modulation>
    <interleaver-depth>fecI8J16</interleaver-depth>
  </docsis-phy-profile>
</chassis>
<docsis>
  <docs-global>
        <maximum-scheduled-codes-enabled>false</maximum-scheduled-codes-enabled>
        <l2-vpn-global-enabled>false</l2-vpn-global-enabled>
  </docs-global>
  <cm-vendor-oui>
    <cm-oui>FFFFFF</cm-oui>
    <cm-vendor-name>CableLabs</cm-vendor-name>
  </cm-vendor-oui>
  <docs-security>
    <sav-config-list>
      <sav-config-list-name>SecCfgSavList1</sav-config-list-name>
      <sav-rule>
        <rule-id>1</rule-id>
        <prefix-address>10.193.1.1/32</prefix-address>
      </sav-rule>
    </sav-config-list>
    <sav-config-list>
      <sav-config-list-name>SecCfgSavList2</sav-config-list-name>
      <sav-rule>
        <rule-id>1</rule-id>
        <prefix-address>10.194.1.1/32</prefix-address>
      </sav-rule>
      <sav-rule>
        <rule-id>2</rule-id>
        <prefix-address>10.194.2.1/32</prefix-address>
      </sav-rule>
    </sav-config-list>
    <cmts-sav-control>
      <cm-authentication-enable>true</cm-authentication-enable>
    </cmts-sav-control>
    <cmts-server-config>
          <tftp-options>net-addr</tftp-options>
          <config-file-learning-enabled>true</config-file-learning-enabled>
    </cmts-server-config>
    <cmts-encrypt>
      <encrypt-alg-priority>aes128CbcMode des56CbcMode des40CbcMode</encrypt-alg-priority>
    </cmts-encrypt>
    <cmts-certificate>
      <cert-revocation-method>crl-and-ocsp</cert-revocation-method>
    </cmts-certificate>
    <cmts-cert-revocation-list>
      <url>crl.verisign.net</url>
      <refresh-interval>10080</refresh-interval>
    </cmts-cert-revocation-list>
    <cmts-cm-eae-exclusion>
      <cmts-cm-eae-exclusion-id>1</cmts-cm-eae-exclusion-id>
      <mac-address>59:94:6B:7C:2A:CC</mac-address>
      <mac-address-mask>FF:FF:FF:FF:FF:FF</mac-address-mask>
    </cmts-cm-eae-exclusion>
    <cmts-online-cert-status-protocol>
      <url>ocsp.verisign.net</url>
      <signature-bypass>false</signature-bypass>
    </cmts-online-cert-status-protocol>
    <sys-bpi-config>
          <sys-default-authentication-lifetime>5</sys-default-authentication-lifetime>
          <sys-default-tek-lifetime>5</sys-default-tek-lifetime>
    </sys-bpi-config>
  </docs-security>
  <docs-subscriber-management>
    <base>
      <cpe-max-ipv4>16</cpe-max-ipv4>
      <cpe-max-ipv6>16</cpe-max-ipv6>
      <cpe-active>true</cpe-active>
      <cpe-learnable>true</cpe-learnable>
      <subscriber-downstream-filter>1</subscriber-downstream-filter>
      <subscriber-upstream-filter>0</subscriber-upstream-filter>
      <cm-downstream-filter>0</cm-downstream-filter>
      <cm-upstream-filter>0</cm-upstream-filter>
      <ps-downstream-filter>0</ps-downstream-filter>
      <ps-upstream-filter>0</ps-upstream-filter>
      <mta-downstream-filter>0</mta-downstream-filter>
```

```
    <mta-upstream-filter>0</mta-upstream-filter>
    <stb-downstream-filter>0</stb-downstream-filter>
    <stb-upstream-filter>0</stb-upstream-filter>
  </base>
  <filter-group>
    <group-id>1</group-id>
    <rule-id>1</rule-id>
    <filter-action>permit</filter-action>
    <priority>1</priority>
    <ip-tos-low>00</ip-tos-low>
    <ip-tos-high>FF</ip-tos-high>
    <ip-tos-mask>FF</ip-tos-mask>
    <ip-protocol>257</ip-protocol>
    <source-address>10.10.10.0/10</source-address>
    <destination-address>192.168.8.1/10</destination-address>
    <source-port-start>16</source-port-start>
    <source-port-end>128</source-port-end>
    <destination-port-start>16</destination-port-start>
    <destination-port-end>128</destination-port-end>
    <destination-mac-address>AA:BB:CC:DD:00:00</destination-mac-address>
    <destination-mac-mask>FF:FF:FF:FF:00:00</destination-mac-mask>
    <source-mac-address>FF:FF:FF:FF:FF:FF</source-mac-address>
    <ethernet-protocol-id>mac</ethernet-protocol-id>
    <ethernet-protocol>0800</ethernet-protocol>
    <user-priority-low>0</user-priority-low>
    <user-priority-high>7</user-priority-high>
    <vlan-id>0</vlan-id>
    <flow-label>0</flow-label>
    <cm-interface-mask>eCm</cm-interface-mask>
  </filter-group>
</docs-subscriber-management>
<docs-qos>
  <service-class>
    <service-class-name>ServiceClass1</service-class-name>
    <priority>0</priority>
    <max-traffic-rate>0</max-traffic-rate>
    <max-traffic-burst>3044</max-traffic-burst>
    <min-reserved-rate>0</min-reserved-rate>
    <min-reserved-packet>12</min-reserved-packet>
    <max-concatenated-burst>1522</max-concatenated-burst>
    <nominal-polling-interval>0</nominal-polling-interval>
    <tolerated-poll-jitter>0</tolerated-poll-jitter>
    <unsolicited-grant-size>0</unsolicited-grant-size>
    <nominal-grant-interval>0</nominal-grant-interval>
    <tolerated-grant-jitter>0</tolerated-grant-jitter>
    <grants-per-interval>0</grants-per-interval>
    <max-latency>0</max-latency>
    <active-timeout>0</active-timeout>
    <admitted-timeout>200</admitted-timeout>
    <scheduling-type>best-effort</scheduling-type>
    <request-policy>00000000</request-policy>
    <tos-and-mask>00</tos-and-mask>
    <tos-or-mask>00</tos-or-mask>
    <direction>upstream</direction>
    <dscp-overwrite>-1</dscp-overwrite>
    <required-attribute-mask>bonded</required-attribute-mask>
    <forbidden-attribute-mask>bonded</forbidden-attribute-mask>
    <attribute-aggregate-rule-mask>00000000</attribute-aggregate-rule-mask>
    <application-id>12</application-id>
    <multiplier-contention-request-window>8</multiplier-contention-request-window>
    <multiplier-bytes-requested>4</multiplier-bytes-requested>
    <max-requests-per-sid-cluster>0</max-requests-per-sid-cluster>
    <max-outstanding-bytes-per-sid-cluster>0</max-outstanding-bytes-per-sid-cluster>
    <max-total-bytes-requested-per-sid-cluster>0</max-total-bytes-requested-per-sid-cluster>
    <max-time-in-sid-cluster>0</max-time-in-sid-cluster>
    <peak-traffic-rate>0</peak-traffic-rate>
    <ds-resequencing>resequencing-dsid</ds-resequencing>
    <minimum-buffer>0</minimum-buffer>
    <target-buffer>0</target-buffer>
    <maximum-buffer>4294967295</maximum-buffer>
  </service-class>
  <qos-profile>
    <qos-profile-index>1</qos-profile-index>
    <priority>0</priority>
    <max-up-bandwidth>0</max-up-bandwidth>
    <guaranteed-up-bandwidth>0</guaranteed-up-bandwidth>
    <max-down-bandwidth>0</max-down-bandwidth>
    <baseline-privacy>false</baseline-privacy>
```

```
        <max-transmit-burst>0</max-transmit-burst>
      </qos-profile>
    </docs-qos>
    <docs-multicast-qos>
      <default-group-service-class>ServiceClass1</default-group-service-class>
      <group-phs-config>
        <group-phs-config-id>1</group-phs-config-id>
        <phs-field>0000</phs-field>
        <phs-mask>0000</phs-mask>
        <phs-size>4</phs-size>
        <phs-verify>true</phs-verify>
      </group-phs-config>
      <group-config>
        <group-config-id>1</group-config-id>
        <rule-priority>1</rule-priority>
        <source-prefix-address>10.10.10.10/32</source-prefix-address>
        <group-prefix-address>231.10.10.10/32</group-prefix-address>
        <tos-low>00</tos-low>
        <tos-high>00</tos-high>
        <tos-mask>FF</tos-mask>
        <group-qos-config-id>1</group-qos-config-id>
        <group-encryption-config-id>1</group-encryption-config-id>
        <group-phs-config-id>1</group-phs-config-id>
      </group-config>
      <group-encryption-config>
        <group-encryption-config-id>1</group-encryption-config-id>
        <control>cmts</control>
        <algorithm>des40-cbc-mode</algorithm>
      </group-encryption-config>
      <group-qos-config>
        <group-qos-config-id>1</group-qos-config-id>
        <service-class-name>ServiceClass1</service-class-name>
        <qos-control>single-session</qos-control>
        <aggregated-session-limit>1600</aggregated-session-limit>
        <application-id>12</application-id>
      </group-qos-config>
    </docs-multicast-qos>
    <docs-mac-domain>
      <downstream-bonding-group>
        <bonding-group-name>1</bonding-group-name>
        <sf-provisioned-attribute-mask>bonded</sf-provisioned-attribute-mask>
        <dsid-resequencing-warning-threshold>255</dsid-resequencing-warning-threshold>
        <dsid-resequencing-wait-time>255</dsid-resequencing-wait-time>
        <docsis-down-channel-ref>
          <slot>1</slot>
          <ds-rf-port>0</ds-rf-port>
          <down-channel>17</down-channel>
        </docsis-down-channel-ref>
        <docsis-down-channel-ref>
          <slot>1</slot>
          <ds-rf-port>0</ds-rf-port>
          <down-channel>18</down-channel>
        </docsis-down-channel-ref>
        <docsis-down-channel-ref>
          <slot>1</slot>
          <ds-rf-port>0</ds-rf-port>
          <down-channel>32</down-channel>
        </docsis-down-channel-ref>
      </downstream-bonding-group>
      <deny-cm>
        <device-mac-address>00:43:21:00:19:73</device-mac-address>
      </deny-cm>
      <deny-cm>
        <device-mac-address>00:43:21:00:99:88</device-mac-address>
      </deny-cm>
      <mac-domain>
        <mac-domain-name>MacDomain1</mac-domain-name>
        <ip-provisioning-mode>ipv4-only</ip-provisioning-mode>
        <admin-state>up</admin-state>
        <up-down-trap-enabled>false</up-down-trap-enabled>
        <mdd-interval>2000</mdd-interval>
        <cm-status-event-control-enabled>true</cm-status-event-control-enabled>
        <upstream-frequency-range>standard</upstream-frequency-range>
        <multicast-dsid-forward-enabled>true</multicast-dsid-forward-enabled>
        <multiple-receive-channel-mode-enabled>true</multiple-receive-channel-mode-enabled>
        <multiple-transmit-channel-mode-enabled>true</multiple-transmit-channel-mode-enabled>
        <early-auth-encrypt-control>enable-eae-ranging-based-enforcement</early-auth-encrypt-
control>
```

```
<tftp-proxy-enabled>true</tftp-proxy-enabled>
<source-address-verification-enabled>true</source-address-verification-enabled>
<cm-udc-enabled>false</cm-udc-enabled>
<send-udc-rules-enabled>false</send-udc-rules-enabled>
<service-type-id-list>00</service-type-id-list>
<bpi2-enforce-control>qosCfgFileWithBpi2Enabled</bpi2-enforce-control>
<md-bpi-config>
  <default-authentication-lifetime>7</default-authentication-lifetime>
  <default-tek-lifetime>7</default-tek-lifetime>
</md-bpi-config>
<upstream-bonding-group>
  <bonding-group-name>UsBondingGroup1</bonding-group-name>
  <sf-provisioned-attribute-mask>bonded</sf-provisioned-attribute-mask>
  <upstream-logical-channel-ref>
    <slot>9</slot>
    <us-rf-port>0</us-rf-port>
    <upstream-physical-channel>0</upstream-physical-channel>
    <upstream-logical-channel>0</upstream-logical-channel>
  </upstream-logical-channel-ref>
  <upstream-logical-channel-ref>
    <slot>9</slot>
    <us-rf-port>0</us-rf-port>
    <upstream-physical-channel>1</upstream-physical-channel>
    <upstream-logical-channel>0</upstream-logical-channel>
  </upstream-logical-channel-ref>
</upstream-bonding-group>
<rcc-configuration>
  <rcp-id>0010000003</rcp-id>
  <rcc-cfg-id>1</rcc-cfg-id>
  <vendor-specific/>
  <description>VendorA</description>
  <receive-channel-configuration>
    <receive-channel-id>1</receive-channel-id>
    <primary-downstream-indicator>true</primary-downstream-indicator>
    <rc-rm-connectivity-identifier>1</rc-rm-connectivity-identifier>
    <docsis-down-channel-ref>
      <slot>1</slot>
      <ds-rf-port>0</ds-rf-port>
      <down-channel>17</down-channel>
    </docsis-down-channel-ref>
  </receive-channel-configuration>
  <receive-channel-configuration>
    <receive-channel-id>2</receive-channel-id>
    <primary-downstream-indicator>true</primary-downstream-indicator>
    <rc-rm-connectivity-identifier>1</rc-rm-connectivity-identifier>
    <docsis-down-channel-ref>
      <slot>1</slot>
      <ds-rf-port>0</ds-rf-port>
      <down-channel>18</down-channel>
    </docsis-down-channel-ref>
  </receive-channel-configuration>
  <receive-channel-configuration>
    <receive-channel-id>3</receive-channel-id>
    <primary-downstream-indicator>false</primary-downstream-indicator>
    <rc-rm-connectivity-identifier>2</rc-rm-connectivity-identifier>
    <docsis-down-channel-ref>
      <slot>1</slot>
      <ds-rf-port>0</ds-rf-port>
      <down-channel>32</down-channel>
    </docsis-down-channel-ref>
  </receive-channel-configuration>
  <receive-module-configuration>
    <receive-module-id>1</receive-module-id>
    <rm-rm-connectivity-id>17</rm-rm-connectivity-id>
    <first-center-frequency>651000000</first-center-frequency>
  </receive-module-configuration>
  <receive-module-configuration>
    <receive-module-id>2</receive-module-id>
    <rm-rm-connectivity-id>17</rm-rm-connectivity-id>
    <first-center-frequency>741000000</first-center-frequency>
  </receive-module-configuration>
  <receive-module-configuration>
    <receive-module-id>17</receive-module-id>
    <rm-rm-connectivity-id>0</rm-rm-connectivity-id>
  </receive-module-configuration>
</rcc-configuration>
<cmts-mac-interface-config>
      <sync-interval>1</sync-interval>
```

```
            <ucd-interval>1</ucd-interval>
            <invited-ranging-attempts>1</invited-ranging-attempts>
            <im-insertion-interval>1</im-insertion-interval>
            <docsis11-concatenation-enabled>true</docsis11-concatenation-enabled>
            <docsis11-fragmentation-enabled>true</docsis11-fragmentation-enabled>
      </cmts-mac-interface-config>
      <upstream-physical-channel-ref>
        <slot>9</slot>
        <us-rf-port>0</us-rf-port>
        <upstream-physical-channel>0</upstream-physical-channel>
      </upstream-physical-channel-ref>
      <upstream-physical-channel-ref>
        <slot>9</slot>
        <us-rf-port>0</us-rf-port>
        <upstream-physical-channel>1</upstream-physical-channel>
      </upstream-physical-channel-ref>
      <upstream-physical-channel-ref>
        <slot>9</slot>
        <us-rf-port>0</us-rf-port>
        <upstream-physical-channel>5</upstream-physical-channel>
      </upstream-physical-channel-ref>
      <non-primary-capable-ds>
        <slot>1</slot>
        <ds-rf-port>0</ds-rf-port>
        <down-channel>32</down-channel>
      </non-primary-capable-ds>
      <primary-capable-ds>
        <slot>1</slot>
        <ds-rf-port>0</ds-rf-port>
        <down-channel>17</down-channel>
      </primary-capable-ds>
      <primary-capable-ds>
        <slot>1</slot>
        <ds-rf-port>0</ds-rf-port>
        <down-channel>18</down-channel>
      </primary-capable-ds>
    </mac-domain>
  </docs-mac-domain>
  <docs-multicast-authorization>
    <control>
      <enable>disable</enable>
      <default-profile-name-list>taglist</default-profile-name-list>
      <default-action>deny</default-action>
      <default-max-number-sessions>0</default-max-number-sessions>
    </control>
    <profiles>
      <mcast-auth-profile-name>AuthProfName1</mcast-auth-profile-name>
      <description>This profile</description>
      <session-rule>
        <session-rule-name>sessionRule1</session-rule-name>
        <id>1</id>
        <priority>0</priority>
        <source-prefix-address>10.10.10.10/32</source-prefix-address>
        <group-prefix-address>10.10.10.10/24</group-prefix-address>
        <authorization-action>deny</authorization-action>
      </session-rule>
    </profiles>
  </docs-multicast-authorization>
  <docs-if>
    <modulation-profile>
      <modulation-index>1</modulation-index>
      <interval-usage-code>
        <usage-code>shortData</usage-code>
        <modulation>qpsk</modulation>
        <preamble-length>3</preamble-length>
        <differential-encoding>false</differential-encoding>
        <fec-error-correction>0</fec-error-correction>
        <fec-codeword-length>32</fec-codeword-length>
        <scrambler-seed>0</scrambler-seed>
        <max-burst-size>4</max-burst-size>
        <last-codeword-shortened>true</last-codeword-shortened>
        <scrambler>false</scrambler>
        <byte-interleaver-depth>1</byte-interleaver-depth>
        <byte-interleaver-block-size>18</byte-interleaver-block-size>
        <preamble>qpsk0</preamble>
        <tcm-error-correction-on>false</tcm-error-correction-on>
        <scdma-interleaver-step-size>1</scdma-interleaver-step-size>
        <scdma-spreader-enable>true</scdma-spreader-enable>
```

```
      <scdma-subframe-codes>1</scdma-subframe-codes>
      <channel-type>tdma</channel-type>
    </interval-usage-code>
  </modulation-profile>
</docs-if>
<docs-packet-cable>
  <packet-cable-config>
        <packet-cable-enabled>true</packet-cable-enabled>
        <pcmm-enabled>true</pcmm-enabled>
        <pc-t0-timer>30</pc-t0-timer>
        <pc-t1-timer>200</pc-t1-timer>
        <pc-t7-timer>200</pc-t7-timer>
        <pc-t8-timer>0</pc-t8-timer>
        <pcmm-t1-timer>200</pcmm-t1-timer>
        <cmts-gate-id-value>47</cmts-gate-id-value>
        <tos>-1</tos>
        <cops-connection-threshold>4000</cops-connection-threshold>
        <control-point-discovery-enabled>true</control-point-discovery-enabled>
  </packet-cable-config>
  <pc-event-config>
        <retry-timer>4000</retry-timer>
        <retry-limit>3</retry-limit>
        <batch-size>5</batch-size>
        <max-age>5</max-age>
        <billing-events>true</billing-events>
  </pc-event-config>
</docs-packet-cable>
<docs-dsg>
  <dsg-timer-config>
    <timer-config-index>1</timer-config-index>
    <init-t-dsg-1>2</init-t-dsg-1>
    <oper-t-dsg-2>600</oper-t-dsg-2>
    <two-way-t-dsg-3>300</two-way-t-dsg-3>
    <one-way-t-dsg-4>1800</one-way-t-dsg-4>
  </dsg-timer-config>
  <dsg-downstream>
    <dsg-downstream-index>7</dsg-downstream-index>
    <enable-dcd>true</enable-dcd>
    <docsis-down-channel-ref>
      <slot>1</slot>
      <ds-rf-port>0</ds-rf-port>
      <down-channel>18</down-channel>
    </docsis-down-channel-ref>
    <timer-config-index>1</timer-config-index>
    <vendor-param-id>1</vendor-param-id>
    <dsg-channel-list-index>44</dsg-channel-list-index>
  </dsg-downstream>
  <dsg-downstream>
    <dsg-downstream-index>8</dsg-downstream-index>
    <enable-dcd>true</enable-dcd>
    <docsis-down-channel-ref>
      <slot>1</slot>
      <ds-rf-port>0</ds-rf-port>
      <down-channel>17</down-channel>
    </docsis-down-channel-ref>
    <timer-config-index>1</timer-config-index>
    <vendor-param-id>2</vendor-param-id>
    <dsg-channel-list-index>44</dsg-channel-list-index>
  </dsg-downstream>
  <dsg-downstream>
    <dsg-downstream-index>9</dsg-downstream-index>
    <enable-dcd>true</enable-dcd>
    <docsis-down-channel-ref>
      <slot>1</slot>
      <ds-rf-port>0</ds-rf-port>
      <down-channel>32</down-channel>
    </docsis-down-channel-ref>
    <timer-config-index>0</timer-config-index>
    <vendor-param-id>0</vendor-param-id>
    <dsg-channel-list-index>44</dsg-channel-list-index>
  </dsg-downstream>
  <dsg-channel-list>
    <dsg-channel-list-index>44</dsg-channel-list-index>
    <dsg-channel>
      <dsg-channel-index>1</dsg-channel-index>
      <channel-downstream-frequency>651000000</channel-downstream-frequency>
    </dsg-channel>
    <dsg-channel>
```

```
    <dsg-channel-index>2</dsg-channel-index>
    <channel-downstream-frequency>657000000</channel-downstream-frequency>
  </dsg-channel>
</dsg-channel-list>
<tunnel-group-to-channel-list>
  <tunnel-group-index>100</tunnel-group-index>
  <tunnel-group-channel>
    <tunnel-group-channel-index>1</tunnel-group-channel-index>
    <rule-priority>0</rule-priority>
    <vendor-param-id>2</vendor-param-id>
    <dsg-downstream-index>7</dsg-downstream-index>
  </tunnel-group-channel>
  <tunnel-group-channel>
    <tunnel-group-channel-index>2</tunnel-group-channel-index>
    <rule-priority>1</rule-priority>
    <vendor-param-id>2</vendor-param-id>
    <dsg-downstream-index>8</dsg-downstream-index>
  </tunnel-group-channel>
</tunnel-group-to-channel-list>
<dsg-tunnel-config>
  <dsg-tunnel-config-index>1</dsg-tunnel-config-index>
  <tunnel-grp-index>100</tunnel-grp-index>
  <mac-address>00:00:00:00:00:00</mac-address>
  <client-id-list-index>1</client-id-list-index>
  <service-class-name>ServiceClass1</service-class-name>
</dsg-tunnel-config>
<dsg-tunnel-config>
  <dsg-tunnel-config-index>2</dsg-tunnel-config-index>
  <tunnel-grp-index>100</tunnel-grp-index>
  <mac-address>00:00:00:00:00:01</mac-address>
  <client-id-list-index>2</client-id-list-index>
  <service-class-name>ServiceClass1</service-class-name>
</dsg-tunnel-config>
<dsg-classifier>
  <dsg-classifier-id>1</dsg-classifier-id>
  <tunnel-index>1</tunnel-index>
  <priority>0</priority>
  <source-ip>10.10.10.11/32</source-ip>
  <destination-ip>231.10.10.11</destination-ip>
  <destination-port-start>0</destination-port-start>
  <destination-port-end>65535</destination-port-end>
  <include-in-dcd>true</include-in-dcd>
</dsg-classifier>
<dsg-classifier>
  <dsg-classifier-id>2</dsg-classifier-id>
  <tunnel-index>1</tunnel-index>
  <priority>1</priority>
  <source-ip>10.10.10.10/32</source-ip>
  <destination-ip>231.10.10.10</destination-ip>
  <destination-port-start>0</destination-port-start>
  <destination-port-end>65535</destination-port-end>
  <include-in-dcd>true</include-in-dcd>
</dsg-classifier>
<dsg-classifier>
  <dsg-classifier-id>3</dsg-classifier-id>
  <tunnel-index>2</tunnel-index>
  <priority>0</priority>
  <source-ip>10.10.10.10/32</source-ip>
  <destination-ip>231.20.20.20</destination-ip>
  <destination-port-start>0</destination-port-start>
  <destination-port-end>65535</destination-port-end>
  <include-in-dcd>true</include-in-dcd>
</dsg-classifier>
<vendor-parameters-list>
  <vendor-param-id>1</vendor-param-id>
  <vendor-param>
    <vendor-index>1</vendor-index>
    <vendor-oui>010203</vendor-oui>
    <vendor-value>0102030405060708090a0b</vendor-value>
  </vendor-param>
  <vendor-param>
    <vendor-index>2</vendor-index>
    <vendor-oui>010203</vendor-oui>
    <vendor-value>0f0e0d0c0b0a</vendor-value>
  </vendor-param>
</vendor-parameters-list>
<vendor-parameters-list>
  <vendor-param-id>2</vendor-param-id>
```

```
    <vendor-param>
      <vendor-index>1</vendor-index>
      <vendor-oui>040506</vendor-oui>
      <vendor-value>1122334455667788899</vendor-value>
    </vendor-param>
  </vendor-parameters-list>
  <client-id-config-list>
    <client-id-list-index>1</client-id-list-index>
    <dsg-client>
      <client-id-index>1</client-id-index>
      <dsg-client-id-type>broadcast</dsg-client-id-type>
      <client-id-value>000000000005</client-id-value>
      <vendor-parameters-id>1</vendor-parameters-id>
    </dsg-client>
    <dsg-client>
      <client-id-index>2</client-id-index>
      <dsg-client-id-type>mac-address</dsg-client-id-type>
      <client-id-value>010203040506</client-id-value>
    </dsg-client>
  </client-id-config-list>
  <client-id-config-list>
    <client-id-list-index>2</client-id-list-index>
    <dsg-client>
      <client-id-index>1</client-id-index>
      <dsg-client-id-type>application-id</dsg-client-id-type>
      <client-id-value>000000000800</client-id-value>
    </dsg-client>
  </client-id-config-list>
</docs-dsg>
<docs-load-balancing>
    <load-balancing-policy>
        <policy-id>1</policy-id>
        <load-balance-rule>
            <rule-id>2</rule-id>
        </load-balance-rule>
    </load-balancing-policy>
    <basic-rule>
        <rule-id>2</rule-id>
        <enable>enabled</enable>
    </basic-rule>
    <general-grp-cfg>
        <mac-domain-name>MacDomain1</mac-domain-name>
        <fiber-node>
            <fiber-node-index>10</fiber-node-index>
        </fiber-node>
        <fiber-node>
            <fiber-node-index>16</fiber-node-index>
        </fiber-node>
        <policy-id>1</policy-id>
    </general-grp-cfg>
    <restricted-grp-cfg>
        <res-grp-id>100</res-grp-id>
        <grp-mac-domain>
            <mac-domain-name>MacDomain1</mac-domain-name>
        </grp-mac-domain>
        <init-tech>reinit-mac</init-tech>
        <policy-id>1</policy-id>
        <upstream-logical-channel-ref>
            <slot>9</slot>
            <us-rf-port>0</us-rf-port>
            <upstream-physical-channel>1</upstream-physical-channel>
            <upstream-logical-channel>0</upstream-logical-channel>
        </upstream-logical-channel-ref>
        <upstream-logical-channel-ref>
            <slot>9</slot>
            <us-rf-port>0</us-rf-port>
            <upstream-physical-channel>0</upstream-physical-channel>
            <upstream-logical-channel>0</upstream-logical-channel>
        </upstream-logical-channel-ref>
        <docsis-down-channel-ref>
            <slot>1</slot>
            <ds-rf-port>0</ds-rf-port>
            <down-channel>17</down-channel>
        </docsis-down-channel-ref>
        <docsis-down-channel-ref>
            <slot>1</slot>
            <ds-rf-port>0</ds-rf-port>
            <down-channel>18</down-channel>
```

```
          </docsis-down-channel-ref>
      </restricted-grp-cfg>
  </docs-load-balancing>
</docsis>
<video>
  <global-input-ts-config>
    <jitter-tolerance>100</jitter-tolerance>
    <unicast-session-loss-timeout>5000</unicast-session-loss-timeout>
    <multicast-session-loss-timeout>5000</multicast-session-loss-timeout>
  </global-input-ts-config>
  <global-output-ts-config>
    <cat-insert-rate>10</cat-insert-rate>
    <pat-insert-rate>10</pat-insert-rate>
    <pmt-insert-rate>10</pmt-insert-rate>
  </global-output-ts-config>
  <video-input-ts>
    <input-ts-index>1</input-ts-index>
    <input-ts-name>CNN</input-ts-name>
    <multicast-video-input-ts>
      <multicast-ts>
        <multicast-ts-source-ip-address>10.0.0.9</multicast-ts-source-ip-address>
        <multicast-ts-destination-ip-address>232.100.0.0</multicast-ts-destination-ip-address>
        <multicast-ts-priority>127</multicast-ts-priority>
      </multicast-ts>
    </multicast-video-input-ts>
  </video-input-ts>
  <video-input-ts>
    <input-ts-index>2</input-ts-index>
    <input-ts-name>ABC</input-ts-name>
    <input-ts-decryption-enabled>true</input-ts-decryption-enabled>
    <multicast-video-input-ts>
      <multicast-ts>
        <multicast-ts-source-ip-address>10.0.0.9</multicast-ts-source-ip-address>
        <multicast-ts-destination-ip-address>232.100.0.1</multicast-ts-destination-ip-address>
        <multicast-ts-priority>127</multicast-ts-priority>
      </multicast-ts>
    </multicast-video-input-ts>
  </video-input-ts>
  <video-input-ts>
    <input-ts-index>3</input-ts-index>
    <input-ts-name>Music </input-ts-name>
    <input-ts-decryption-enabled>true</input-ts-decryption-enabled>
    <multicast-video-input-ts>
      <multicast-ts>
        <multicast-ts-source-ip-address>10.0.0.9</multicast-ts-source-ip-address>
        <multicast-ts-destination-ip-address>232.100.1.1</multicast-ts-destination-ip-address>
        <multicast-ts-destination-udp-port>2000</multicast-ts-destination-udp-port>
        <multicast-ts-priority>100</multicast-ts-priority>
      </multicast-ts>
      <multicast-ts>
        <multicast-ts-source-ip-address>10.0.0.10</multicast-ts-source-ip-address>
        <multicast-ts-destination-ip-address>232.100.1.1</multicast-ts-destination-ip-address>
        <multicast-ts-destination-udp-port>2000</multicast-ts-destination-udp-port>
        <multicast-ts-priority>50</multicast-ts-priority>
      </multicast-ts>
    </multicast-video-input-ts>
  </video-input-ts>
  <video-input-ts>
    <input-ts-index>4</input-ts-index>
    <input-ts-name>HBO</input-ts-name>
    <multicast-video-input-ts>
      <multicast-ts>
        <multicast-ts-source-ip-address>10.0.0.9</multicast-ts-source-ip-address>
        <multicast-ts-destination-ip-address>232.100.1.100</multicast-ts-destination-ip-address>
        <multicast-ts-destination-udp-port>2000</multicast-ts-destination-udp-port>
        <multicast-ts-priority>100</multicast-ts-priority>
      </multicast-ts>
      <multicast-ts>
        <multicast-ts-source-ip-address>10.0.0.10</multicast-ts-source-ip-address>
        <multicast-ts-destination-ip-address>232.100.1.100</multicast-ts-destination-ip-address>
        <multicast-ts-destination-udp-port>2000</multicast-ts-destination-udp-port>
        <multicast-ts-priority>100</multicast-ts-priority>
      </multicast-ts>
    </multicast-video-input-ts>
  </video-input-ts>
  <video-input-ts>
    <input-ts-index>5</input-ts-index>
    <unicast-video-input-ts>
```

```
    <address>
        <unicast-ts-destination-ip-address>10.10.10.99</unicast-ts-destination-ip-address>
    </address>
    <unicast-ts-destination-udp-port>2000</unicast-ts-destination-udp-port>
  </unicast-video-input-ts>
</video-input-ts>
<video-input-ts>
  <input-ts-index>6</input-ts-index>
  <unicast-video-input-ts>
    <interface>
        <unicast-ts-interface-name>eth6/0</unicast-ts-interface-name>
    </interface>
    <unicast-ts-destination-udp-port>3000</unicast-ts-destination-udp-port>
  </unicast-video-input-ts>
</video-input-ts>
<static-udp-map>
  <udp-map-index>0</udp-map-index>
  <starting-udp-port>5000</starting-udp-port>
  <port-count>10</port-count>
  <static-video-output-ts>0</static-video-output-ts>
</static-udp-map>
<static-udp-map>
  <udp-map-index>1</udp-map-index>
  <starting-udp-port>5010</starting-udp-port>
  <port-count>10</port-count>
  <static-video-output-ts>1</static-video-output-ts>
</static-udp-map>
<reserved-udp-map>
  <udp-map-index>0</udp-map-index>
  <starting-udp-port>0</starting-udp-port>
  <port-count>1024</port-count>
</reserved-udp-map>
<reserved-pid-range>
  <reserved-pid-range-index>0</reserved-pid-range-index>
  <starting-pid>0</starting-pid>
  <count>32</count>
  <description>MPEG-2 and DVB reserved</description>
</reserved-pid-range>
<reserved-pid-range>
  <reserved-pid-range-index>1</reserved-pid-range-index>
  <starting-pid>32</starting-pid>
  <count>224</count>
  <description>Reserved for non-remapped pid session</description>
</reserved-pid-range>
<reserved-pid-range>
  <reserved-pid-range-index>2</reserved-pid-range-index>
  <starting-pid>8187</starting-pid>
  <count>5</count>
  <description>MPEG-2 and ATSC reserved</description>
</reserved-pid-range>
<input-registration>
  <input-registration-name>eth6/0</input-registration-name>
  <group-name>EDGE-IN-GROUP-1</group-name>
  <erm-name>ERM-1</erm-name>
  <bandwidth>0</bandwidth>
  <erm-managed-input>true</erm-managed-input>
</input-registration>
<input-registration>
  <input-registration-name>eth6/3a</input-registration-name>
  <group-name>GROUP-1A</group-name>
  <erm-name>ERM-1</erm-name>
  <bandwidth>1000000</bandwidth>
  <erm-managed-input>true</erm-managed-input>
</input-registration>
<input-registration>
  <input-registration-name>eth6/3b</input-registration-name>
  <group-name>GROUP-1B</group-name>
  <erm-name>ERM-1</erm-name>
  <bandwidth>1000000</bandwidth>
  <erm-managed-input>true</erm-managed-input>
</input-registration>
<input-registration>
  <input-registration-name>eth7/0</input-registration-name>
  <group-name>EDGE-IN-GROUP-1</group-name>
  <erm-name>ERM-1</erm-name>
  <bandwidth>0</bandwidth>
  <erm-managed-input>true</erm-managed-input>
</input-registration>
```

```
<input-registration>
  <input-registration-name>eth7/6</input-registration-name>
  <group-name>GROUP-2A</group-name>
  <erm-name>ERM-2</erm-name>
  <bandwidth>1000000</bandwidth>
  <erm-managed-input>true</erm-managed-input>
</input-registration>
<pid-session>
  <session-index>100</session-index>
  <session-name>HBO</session-name>
  <session-input-ts>4</session-input-ts>
  <session-output-ts>
    <session-output-ts-index>0</session-output-ts-index>
  </session-output-ts>
  <input-pid>100</input-pid>
  <pid-remap-enable>false</pid-remap-enable>
  <pid-type>pat</pid-type>
  <cas-id>00000000</cas-id>
  <output-pid>1100</output-pid>
</pid-session>
<pid-session>
  <session-index>92</session-index>
  <session-input-ts>5</session-input-ts>
  <session-output-ts>
    <session-output-ts-index>0</session-output-ts-index>
  </session-output-ts>
  <input-pid>92</input-pid>
  <pid-remap-enable>false</pid-remap-enable>
  <pid-type>pat</pid-type>
  <cas-id>00000000</cas-id>
  <output-pid>1092</output-pid>
</pid-session>
<program-session>
  <session-index>0</session-index>
  <session-name>CNN-HD</session-name>
  <session-input-ts>1</session-input-ts>
  <session-output-ts>
    <session-output-ts-index>0</session-output-ts-index>
  </session-output-ts>
  <input-mpeg-program-number>8</input-mpeg-program-number>
  <output-mpeg-program-number>4</output-mpeg-program-number>
  <pat-pid-remap>true</pat-pid-remap>
  <requested-bandwidth>12000000</requested-bandwidth>
  <cas-info>0</cas-info>
  <encryption-data>0</encryption-data>
  <encrypt-control>0</encrypt-control>
</program-session>
<program-session>
  <session-index>1</session-index>
  <session-name>ABC-HD</session-name>
  <session-input-ts>2</session-input-ts>
  <session-output-ts>
    <session-output-ts-index>0</session-output-ts-index>
  </session-output-ts>
  <input-mpeg-program-number>3</input-mpeg-program-number>
  <output-mpeg-program-number>7</output-mpeg-program-number>
  <pat-pid-remap>true</pat-pid-remap>
  <requested-bandwidth>12000000</requested-bandwidth>
  <cas-info>0</cas-info>
  <encryption-data>1</encryption-data>
  <encrypt-control>0</encrypt-control>
</program-session>
<cas-info>
  <cas-info-index>0</cas-info-index>
  <cas-id>00000000</cas-id>
  <ca-blob>String</ca-blob>
</cas-info>
<mpts-passthrough-session>
  <session-index>0</session-index>
  <session-name>Music-channels</session-name>
  <session-input-ts>3</session-input-ts>
  <session-output-ts>
    <session-output-ts-index>1</session-output-ts-index>
  </session-output-ts>
</mpts-passthrough-session>
<encryption-data>
  <encryption-data-index>0</encryption-data-index>
  <cci-level>copy-never</cci-level>
```

```
  <cit>clear</cit>
  <rct>not-asserted</rct>
  <cci-reserved>0</cci-reserved>
  <provider-asset-id>67343-CNN-HD</provider-asset-id>
</encryption-data>
<encryption-data>
  <encryption-data-index>1</encryption-data-index>
  <cci-level>copy-never</cci-level>
  <cit>set</cit>
  <rct>required</rct>
  <cci-reserved>0</cci-reserved>
  <provider-asset-id>89643-ABC-HD</provider-asset-id>
</encryption-data>
<encrypt-control>
  <encrypt-control-index>0</encrypt-control-index>
  <encryption-scheme>dvbcsa</encryption-scheme>
  <block-stream-until-encrypted>true</block-stream-until-encrypted>
  <key-length>128bits</key-length>
  <encryptor-opaque>CA-KEY-0957723545635</encryptor-opaque>
</encrypt-control>
<ecmd>
  <ecm-index>1</ecm-index>
  <ecm-server>
   <address>
    <address>10.0.0.1</address>
   </address>
  </ecm-server>
  <ecm-server-port>65535</ecm-server-port>
  <ecm-cas-id>00000001</ecm-cas-id>
  <number-decrypted-streams>128</number-decrypted-streams>
</ecmd>
<ecmg>
  <ecm-index>1</ecm-index>
  <ecm-server>
   <address>
    <address>10.0.0.1</address>
   </address>
  </ecm-server>
  <ecm-server-port>65535</ecm-server-port>
  <ecm-cas-id>00000001</ecm-cas-id>
  <recommended-cp-duration>5</recommended-cp-duration>
  <number-encrypted-streams>128</number-encrypted-streams>
</ecmg>
<erm-registration>
  <erm-name>ERM-1</erm-name>
  <erm-address>
   <address>
    <address>192.168.0.45</address>
   </address>
  </erm-address>
  <erm-port>6069</erm-port>
  <erm-connection-mode>server</erm-connection-mode>
  <hold-timer>240</hold-timer>
  <connection-retry-timer>120</connection-retry-timer>
  <next-hop-address-domain>0</next-hop-address-domain>
  <comp-address>
   <name>
    <name>google.com</name>
   </name>
  </comp-address>
  <streaming-zone>Zone1</streaming-zone>
  <id>0</id>
  <cost>0</cost>
  <comp-name>Region1,Local2</comp-name>
</erm-registration>
<video-output-ts>
  <output-ts-index>0</output-ts-index>
  <output-ts-name>SDV1</output-ts-name>
  <video-down-channel-ref>
    <slot>1</slot>
    <ds-rf-port>0</ds-rf-port>
    <down-channel>1</down-channel>
  </video-down-channel-ref>
  <video-down-channel-ref>
    <slot>1</slot>
    <ds-rf-port>1</ds-rf-port>
    <down-channel>1</down-channel>
  </video-down-channel-ref>
```

```
      <video-down-channel-ref>
        <slot>3</slot>
        <ds-rf-port>0</ds-rf-port>
        <down-channel>1</down-channel>
      </video-down-channel-ref>
      <video-down-channel-ref>
        <slot>3</slot>
        <ds-rf-port>1</ds-rf-port>
        <down-channel>1</down-channel>
      </video-down-channel-ref>
    </video-output-ts>
    <video-output-ts>
      <output-ts-index>1</output-ts-index>
      <output-ts-name>Music Channels</output-ts-name>
      <video-down-channel-ref>
        <slot>1</slot>
        <ds-rf-port>0</ds-rf-port>
        <down-channel>2</down-channel>
      </video-down-channel-ref>
      <video-down-channel-ref>
        <slot>1</slot>
        <ds-rf-port>1</ds-rf-port>
        <down-channel>2</down-channel>
      </video-down-channel-ref>
      <video-down-channel-ref>
        <slot>3</slot>
        <ds-rf-port>0</ds-rf-port>
        <down-channel>2</down-channel>
      </video-down-channel-ref>
      <video-down-channel-ref>
        <slot>3</slot>
        <ds-rf-port>1</ds-rf-port>
        <down-channel>2</down-channel>
      </video-down-channel-ref>
    </video-output-ts>
    <static-udp-map-encryption>
        <udp-map-encryption-index>2</udp-map-encryption-index>
        <cas-info>0</cas-info>
        <encryption-data>0</encryption-data>
        <encrypt-control>1</encrypt-control>
    </static-udp-map-encryption>
  </video>
  <epon>
    <oam-config>
      <min-oam-rate>1</min-oam-rate>
      <max-oam-rate>30</max-oam-rate>
      <oam-response-timeout>1</oam-response-timeout>
    </oam-config>
    <loop-timing-config>
      <min-propagation-delay>0</min-propagation-delay>
      <max-propagation-delay>6250</max-propagation-delay>
      <onu-delay>3125</onu-delay>
    </loop-timing-config>
    <mpcp-config>
      <discovery-period>700</discovery-period>
      <grant-size-in-discovery-gate>16319</grant-size-in-discovery-gate>
      <deregistration-timeout>0</deregistration-timeout>
    </mpcp-config>
    <deny-onu>
      <onu-mac-address>00:63:44:00:11:29</onu-mac-address>
    </deny-onu>
  </epon>
  <network>
    <dns-resolver>
      <domain-suffix>example.com</domain-suffix>
      <enabled>true</enabled>
    </dns-resolver>
    <dns-server>
      <dns-server-index>1</dns-server-index>
      <server-ip>10.10.10.10</server-ip>
    </dns-server>
    <integrated-servers>
      <server-type>ssh</server-type>
      <local-listener-port>22</local-listener-port>
      <enabled>true</enabled>
      <listener-ip-interface-name>eth0</listener-ip-interface-name>
    </integrated-servers>
    <authentication-policy>
```

```
  <policy>login</policy>
  <protocol>radius</protocol>
  <priority>2</priority>
</authentication-policy>
<local-authorization>
  <username>admin</username>
  <privilege-level>2</privilege-level>
  <password>root</password>
  <clear-key>true</clear-key>
</local-authorization>
<radius>
  <auth-server-index>1</auth-server-index>
  <auth-server>
   <address>
    <address>10.10.10.10</address>
   </address>
  </auth-server>
  <auth-key>testing</auth-key>
  <auth-clear-key>true</auth-clear-key>
  <auth-timeout>3</auth-timeout>
  <auth-retransmit-attempts>1</auth-retransmit-attempts>
  <primary-auth-server>true</primary-auth-server>
  <source-ip-interface-name>eth0</source-ip-interface-name>
  <radius-auth-port>1812</radius-auth-port>
  <accounting-port>1813</accounting-port>
</radius>
<tacacs-plus>
  <auth-server-index>1</auth-server-index>
  <auth-server>
   <address>
    <address>10.10.10.10</address>
   </address>
  </auth-server>
  <auth-key>testing</auth-key>
  <auth-clear-key>true</auth-clear-key>
  <auth-timeout>3</auth-timeout>
  <auth-retransmit-attempts>1</auth-retransmit-attempts>
  <primary-auth-server>true</primary-auth-server>
  <source-ip-interface-name>eth0</source-ip-interface-name>
  <tacacs-plus-auth-port>49</tacacs-plus-auth-port>
</tacacs-plus>
<keychain>
  <key-id>1</key-id>
  <key-string>testing</key-string>
  <accept-lifetime>1000</accept-lifetime>
  <send-lifetime>10000</send-lifetime>
  <clear-key>true</clear-key>
</keychain>
<fail-over>
  <auto-fail-back>true</auto-fail-back>
</fail-over>
<local-time>
  <ntp-master>
   <name>
    <name>time.nist.gov</name>
   </name>
  </ntp-master>
  <time-zone>-07</time-zone>
  <dst-recurring-change>true</dst-recurring-change>
  <source-ip-interface-name>eth0</source-ip-interface-name>
</local-time>
<acl>
    <acl-name>acl1</acl-name>
    <ip-acl-rule>
        <acl-rule-index>1</acl-rule-index>
        <is-rule>
            <acl-action>accept</acl-action>
            <ipv4>
                <ipv4-rule>
                    <dest-ipv4-addr-filter>
                        <dest-addr>10.30.50.0</dest-addr>
                        <dest-wildcard-mask>0.0.0.255</dest-wildcard-mask>
                    </dest-ipv4-addr-filter>
                </ipv4-rule>
            </ipv4>
        </is-rule>
    </ip-acl-rule>
    <ip-acl-rule>
```

```
            <acl-rule-index>2</acl-rule-index>
            <is-rule>
                <acl-action>accept</acl-action>
                <ipv4>
                    <ipv4-rule>
                        <source-ipv4-addr-filter>
                            <source-addr>66.77.88.100</source-addr>
                            <source-wildcard-mask>0.0.0.0</source-wildcard-mask>
                        </source-ipv4-addr-filter>
                        <single-source-port>
                                <sport>1024</sport>
                                <sport-comparator>lt</sport-comparator>
                        </single-source-port>
                    </ipv4-rule>
                </ipv4>
            </is-rule>
        </ip-acl-rule>
    </acl>
    <acl>
        <acl-name>acl2</acl-name>
        <ip-acl-rule>
            <acl-rule-index>1</acl-rule-index>
            <is-rule>
                <acl-action>accept</acl-action>
                <ipv6>
                    <ipv6-rule>
                        <dest-ipv6-addr-filter>
                            <dest-addr>fc00:0:c416:c015::</dest-addr>
                            <dest-wildcard-mask>0000:ffff:0000:0000:ffff:ffff:ffff:ffff</dest-
wildcard-mask>
                        </dest-ipv6-addr-filter>
                        <protocol-value>
                                <protocol-id>6</protocol-id>
                        </protocol-value>
                        <dest-portrange>
                                <start-dport>10000</start-dport>
                                <end-dport>10100</end-dport>
                        </dest-portrange>
                    </ipv6-rule>
                </ipv6>
            </is-rule>
        </ip-acl-rule>
        <ip-acl-rule>
            <acl-rule-index>2</acl-rule-index>
            <is-remark>
                <remark>IPv6 rule</remark>
            </is-remark>
        </ip-acl-rule>
        <ip-acl-rule>
            <acl-rule-index>3</acl-rule-index>
            <is-rule>
                <acl-action>deny</acl-action>
                <ipv6></ipv6>
            </is-rule>
        </ip-acl-rule>
    </acl>
  </network>
  <interface>
    <cable-bundle>
      <interface-index>1</interface-index>
      <admin-state>up</admin-state>
      <ip-interface>
        <ip-interface-name>mac1</ip-interface-name>
        <primary-ipv4>
          <ip-address>192.168.11.7/10</ip-address>
        </primary-ipv4>
        <ipv6>
          <ipv6-address>fe80:0:230:48ff:fe23:4177/10</ipv6-address>
        </ipv6>
        <secondary-ipv4>
          <ip-address>192.168.11.12/10</ip-address>
        </secondary-ipv4>
      </ip-interface>
      <dhcp-giaddr-primary>192.168.11.7</dhcp-giaddr-primary>
      <secondary-giaddr>
        <dhcp-giaddr-secondary>192.168.11.12</dhcp-giaddr-secondary>
      </secondary-giaddr>
      <docs-md>
```

```
      <docsis-mac-domain>
        <docsis-mac-domain-name>MacDomain1</docsis-mac-domain-name>
      </docsis-mac-domain>
    </docs-md>
    <cable-helper-config>
      <cable-helper-config-index>1</cable-helper-config-index>
      <cable-helper-address>
       <address>
         <address>192.168.11.1</address>
       </address>
      </cable-helper-address>
      <application>all</application>
    </cable-helper-config>
    <ingress-acl>acl2</ingress-acl>
    <egress-acl>acl1</egress-acl>
  </cable-bundle>
  <loopback>
    <interface-index>1</interface-index>
    <admin-state>up</admin-state>
    <ip-interface>
      <ip-interface-name>lo</ip-interface-name>
      <primary-ipv4>
        <ip-address>127.0.0.1/32</ip-address>
      </primary-ipv4>
      <ipv6>
        <ipv6-address>fe80:0:230:48ff:fe23:4177/10</ipv6-address>
      </ipv6>
      <secondary-ipv4>
        <ip-address>127.0.0.1/10</ip-address>
      </secondary-ipv4>
    </ip-interface>
  </loopback>
  <mgmd-router-interface>
    <query-interval>125</query-interval>
    <version>igmp-v2-or-mld-v1</version>
    <query-max-response-time>100</query-max-response-time>
    <robustness>4</robustness>
    <last-member-query-interval>25</last-member-query-interval>
  </mgmd-router-interface>
</interface>
<management>
  <ipdr>
    <exporter-config>
      <enabled>true</enabled>
    </exporter-config>
    <streaming-session>
      <session-id>1</session-id>
      <keep-alive-interval>20</keep-alive-interval>
      <ack-time-interval>30</ack-time-interval>
      <ack-sequence-interval>200</ack-sequence-interval>
      <collection-interval>15</collection-interval>
      <streaming-type>time-interval</streaming-type>
      <enabled>true</enabled>
      <service-definition-template>
        <service-definition-id>samis-type-2</service-definition-id>
      </service-definition-template>
      <service-definition-template>
        <service-definition-id>cpe-type</service-definition-id>
      </service-definition-template>
      <collector-reference>
        <collector-id>1</collector-id>
      </collector-reference>
    </streaming-session>
    <collector>
      <collector-id>1</collector-id>
      <collector-ip>10.10.10.10</collector-ip>
      <collector-name>Collector1</collector-name>
      <collector-port>4737</collector-port>
      <priority>1</priority>
    </collector>
  </ipdr>
  <fault-management>
    <event-throttle-config>
      <throttle-admin-state>unconstrained</throttle-admin-state>
      <threshold>50</threshold>
      <interval>1</interval>
    </event-throttle-config>
    <event-reporting-config>
```

```
    <priority>emergency</priority>
    <reporting>local traps syslog</reporting>
  </event-reporting-config>
  <event-reporting-config>
    <priority>alert</priority>
    <reporting>local traps syslog</reporting>
  </event-reporting-config>
  <event-reporting-config>
    <priority>critical</priority>
    <reporting>local traps syslog</reporting>
  </event-reporting-config>
  <event-reporting-config>
    <priority>error</priority>
    <reporting>local syslog</reporting>
  </event-reporting-config>
  <event-reporting-config>
    <priority>warning</priority>
    <reporting>local syslog</reporting>
  </event-reporting-config>
  <event-reporting-config>
    <priority>notice</priority>
    <reporting>local syslog</reporting>
  </event-reporting-config>
  <event-reporting-config>
    <priority>information</priority>
    <reporting>local</reporting>
  </event-reporting-config>
  <event-reporting-config>
    <priority>debug</priority>
    <reporting>local</reporting>
  </event-reporting-config>
  <cmts-event-ctrl>
    <event-id>0</event-id>
  </cmts-event-ctrl>
  <trap-enable>
    <snmp-enable-authen-traps>true</snmp-enable-authen-traps>
  </trap-enable>
  <interface-trap-enable>
    <if-name>IF-1/1</if-name>
    <link-up-down-trap-enable>true</link-up-down-trap-enable>
  </interface-trap-enable>
  <interface-trap-enable>
    <if-name>IF-1/2</if-name>
    <link-up-down-trap-enable>false</link-up-down-trap-enable>
  </interface-trap-enable>
  <syslog-server-config>
    <syslog-server-config-index>0</syslog-server-config-index>
    <syslog-server>
     <address>
      <address>192.168.0.45</address>
     </address>
    </syslog-server>
    <enabled>true</enabled>
  </syslog-server-config>
  <diag-log-triggers-config>
    <include-triggers>ranging-retry</include-triggers>
    <enable-aging-triggers>ranging-retry</enable-aging-triggers>
    <reg-time-interval>90</reg-time-interval>
    <reg-detail>config-file-download-complete</reg-detail>
    <ranging-retry-trigger>consecutive-miss</ranging-retry-trigger>
    <ranging-retry-threshold>6</ranging-retry-threshold>
    <ranging-retry-station-maint-num>90</ranging-retry-station-maint-num>
  </diag-log-triggers-config>
  <diag-log-global-config>
    <max-size>100</max-size>
    <notify-log-size-high-thrshld>80</notify-log-size-high-thrshld>
    <notify-log-size-low-thrshld>60</notify-log-size-low-thrshld>
    <aging>10080</aging>
    <notif-ctrl>high-threshold-reached</notif-ctrl>
  </diag-log-global-config>
</fault-management>
<snmp>
  <access-config>
    <community>public</community>
    <ip-address>192.168.0.20/24</ip-address>
    <type>read-only</type>
    <view-config-ref>
      <view-name>ALL-MIB</view-name>
```

```
        </view-config-ref>
      </access-config>
      <access-config>
        <community>public-v1</community>
        <ip-address>192.168.0.20/24</ip-address>
        <type>read-only</type>
        <view-config-ref>
          <view-name>ALL-MIB</view-name>
        </view-config-ref>
        <view-config-ref>
          <view-name>NO-V2MIB</view-name>
        </view-config-ref>
      </access-config>
      <access-config>
        <community>private</community>
        <ip-address>192.168.0.20/24</ip-address>
        <type>read-write</type>
        <view-config-ref>
          <view-name>ALL-MIB</view-name>
        </view-config-ref>
      </access-config>
      <view-config>
        <view-name>ALL-MIB</view-name>
        <subtree>1</subtree>
        <subtree-mask>0</subtree-mask>
        <type>included</type>
      </view-config>
      <view-config>
        <view-name>NO-V2MIB</view-name>
        <subtree>1.3.6.1.6</subtree>
        <subtree-mask>65535</subtree-mask>
        <type>excluded</type>
      </view-config>
      <notification-receiver-config>
        <notification-receiver-name>NMS-1</notification-receiver-name>
        <type>snmpv2c-inform</type>
        <notification-receiver>
         <address>
          <address>192.168.0.89</address>
         </address>
        </notification-receiver>
        <notification-receiver-port>162</notification-receiver-port>
        <timeout>1</timeout>
        <retries>3</retries>
        <view-config-ref>
          <view-name>ALL-MIB</view-name>
        </view-config-ref>
      </notification-receiver-config>
      <notification-receiver-config>
        <notification-receiver-name>NMS-2</notification-receiver-name>
        <type>snmpv2c-inform</type>
        <notification-receiver>
            <name>
                <name>snmpHost.mso</name>
            </name>
        </notification-receiver>
        <notification-receiver-port>162</notification-receiver-port>
        <timeout>1</timeout>
        <retries>3</retries>
        <view-config-ref>
          <view-name>ALL-MIB</view-name>
        </view-config-ref>
        <view-config-ref>
          <view-name>NO-V2MIB</view-name>
        </view-config-ref>
      </notification-receiver-config>
    </snmp>
  </management>
</ccap:ccap>
```

H.2 CCAP Partial Configuration

See clause 6.3.5 XML Configuration File Execution Command and NETCONF Operations, for sample partial configuration XML files.

H.3 Sample NETCONF Message Exchanges

The following sections show examples of how messages flow between a NETCONF client and the NETCONF server on the CCAP. In the first example, the changes are communicated, but the configuration is not locked. In the second example, the NETCONF client locks the configuration while the session is active. While the session is locked, other users are unable to make changes. If the CCAP is unable to "promote" the candidate configuration to running-config before the timeout period, the changes will be rolled back.

H.3.1 Changes Made to running-config without Locks or Timeouts

In this example, changes are made directly to the running-config. No timeout is set, so the Client waits until the CCAP completes the configuration change.

NETCONF Client and the CCAP send <hello> messages and the CCAP advertises support for its supported version of NETCONF and of the CCAP configuration modules.

```
Client: <hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
Client: <capabilities>
Client: <capability>urn:ietf:params:netconf:base:1.0</capability>
Client: </capabilities>
Client: </hello>

CCAP: <hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
CCAP: <capabilities>
CCAP: <capability>
CCAP: urn:ietf:params:xml:ns:netconf:base:1.0
CCAP: </capability>
CCAP: <capability>
CCAP: urn:cablelabs:params:xml:ns:yang:ccap?revision=2012-08-09?module=ccap
CCAP: </capability>
CCAP: <session-id>101</session-id>
CCAP: </capabilities>
CCAP: </hello>
```

The client successfully updates the running-config with the updated name, description, and location parameters and EPON parameters. The change takes effect immediately.

```
Client: <rpc message-id="1"
Client xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
Client: <edit-config>
Client: <target>
Client: <running/>
Client: </target>
Client: <config>
Client: <ccap xmlns="urn:cablelabs:params:xml:ns:yang:ccap">
Client: <epon>
Client: <oam-config>
Client: <min-oam-rate>2</min-oam-rate>
Client: <max-oam-rate>31</max-oam-rate>
Client: <oam-response-timeout>2</oam-response-timeout>
Client: </oam-config>
Client: <loop-timing-config>
Client: <min-propagation-delay>1</min-propagation-delay>
Client: <max-propagation-delay>6251</max-propagation-delay>
Client: <onu-delay>3126</onu-delay>
Client: </loop-timing-config>
Client: <mpcp-config>
Client: <discovery-period>1001</discovery-period>
Client: <grant-size-in-discovery-gate>16320</grant-size-in-discovery-gate>
Client: <deregistration-timeout>1</deregistration-timeout>
Client: </mpcp-config>
Client: </epon>
Client: </ccap>
Client: </config>
Client: </edit-config>
Client: </rpc>

CCAP: <rpc-reply message-id="1" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
CCAP: <ok/>
CCAP: </rpc-reply>
```

The CCAP copies the running-config to the startup-config.

```
Client: <rpc message-id="2" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
Client: <copy-config>
Client: <target>
Client: <startup/>
Client: </target>
Client: <source>
Client: </running>
Client: </source>
Client: </copy-config>
Client: </rpc>

CCAP: <rpc-reply message-id="2" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
CCAP: <ok/>
CCAP: </rpc-reply>
```

The client then closes the session by sending the <close-session> operation.

```
Client: <rpc message-id="3" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
Client:   <close-session/>
Client: </rpc>
```

The CCAP acknowledges the request and the transport session is subsequently terminated.

```
CCAP: <rpc-reply message-id="3" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
CCAP:   <ok/>
CCAP: </rpc-reply>
```

H.3.2 Changes Made to candidate-config with a Lock

In this example, the Client makes updates to a candidate-config, then instructs the CCAP to copy it to the running-config. If the CCAP is unable to complete this task by the timeout set, then the changes will be rolled back.

NETCONF Client and the CCAP send <hello> messages and the CCAP advertises support for its supported version of NETCONF and of the CCAP configuration modules.

```
Client: <hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
Client: <capabilities>
Client: <capability>urn:ietf:params:netconf:base:1.0</capability>
Client: </capabilities>
Client: </hello>

CCAP: <hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
CCAP: <capabilities>
CCAP: <capability>
CCAP: urn:ietf:params:xml:ns:netconf:base:1.0
CCAP: </capability>
CCAP: <capability>
CCAP: urn:cablelabs:params:xml:ns:yang:ccap?revision=2012-08-09?module=ccap
CCAP: </capability>
CCAP: <session-id>101</session-id>
CCAP: </capabilities>
CCAP: </hello>
```

Client takes a lock on the running datastore.

```
Client: <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
Client:   <lock>
Client:     <target>
Client:       <running/>
Client:     </target>
Client:   </lock>
Client: </rpc>

CCAP: <rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
CCAP:   <ok/>
CCAP: </rpc-reply>
```

The Client successfully updates the candidate-config with the changes to the CCAP parameters and EPON parameters.

```
Client: <rpc message-id="2" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
Client: <edit-config>
Client: <target>
Client: <candidate/>
Client: </target>
Client: <config>
Client: <ccap xmlns="urn:cablelabs:params:xml:ns:yang:ccap">
Client: <epon>
Client: <oam-config>
Client: <min-oam-rate>2</min-oam-rate>
Client: <max-oam-rate>31</max-oam-rate>
Client: <oam-response-timeout>2</oam-response-timeout>
Client: </oam-config>
Client: <loop-timing-config>
Client: <min-propagation-delay>1</min-propagation-delay>
Client: <max-propagation-delay>6251</max-propagation-delay>
Client: <onu-delay>3126</onu-delay>
Client: </loop-timing-config>
Client: <mpcp-config>
Client: <discovery-period>1001</discovery-period>
Client: <grant-size-in-discovery-gate>16320</grant-size-in-discovery-gate>
Client: <deregistration-timeout>1</deregistration-timeout>
Client: </mpcp-config>
Client: </epon>
Client: </ccap>
Client: </config>
Client: </edit-config>
Client: </rpc>

CCAP: <rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="2">
CCAP:   <ok/>
CCAP: </rpc-reply>
```

The Client commits the configuration in the candidate-config to the running-config. This is done with a timeout of 120 seconds. The CCAP is expected to come back with a confirming commit before the timeout expires, otherwise the configuration change will roll back.

```
Client: <rpc message-id="3" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
Client:     <commit>
Client:       <confirmed/>
Client:       <confirm-timeout>120</confirm-timeout>
Client:     </commit>
Client:   </rpc>

CCAP: <rpc-reply message-id="3" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
CCAP:   <ok/>
CCAP: </rpc-reply>
```

The Client does any external tests required and then comes back with a confirming commit.

```
Client: <rpc message-id="4" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
Client:   <commit/>
Client: </rpc>

CCAP: <rpc-reply message-id="4" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
CCAP:   <ok/>
CCAP: </rpc-reply>
```

The Client releases the lock on the running data store allowing other applications to access the configuration.

```
Client: <rpc message-id="5" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
Client:   <unlock>
Client:     <target>
Client:       <running/>
Client:     </target>
Client:   </unlock>
Client: </rpc>

CCAP: <rpc-reply message-id="5" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
CCAP:   <ok/>
CCAP: </rpc-reply>
```

The Client then closes the session by sending the <close-session> operation.

```
Client: <rpc message-id="6" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
Client:   <close-session/>
Client: </rpc>
```

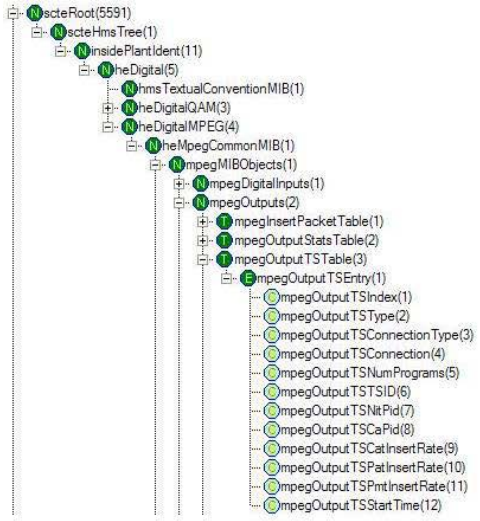
The CCAP acknowledges the request and the transport session is subsequently terminated.

```
CCAP: <rpc-reply message-id="6" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
CCAP:   <ok/>
CCAP: </rpc-reply>
```

Annex I (informative): Use Cases

I.1 Identifying Replicated QAMs

A replicated QAM can be identified by looking at the data presented in the SCTE-HMS-MPEG-MIB. In the mpegOutputTsTable the replicated QAM can be identified by locating instances that have the same mpegOutputTSTSID values. In figure I.1 QAM instance 100 and 137 are replicated - they both have a mpegOutputTSTSID of 1000.



	Instance	mpegOutputTSIndex	mpegOutputTSType	mpegOutputTSConnectionType	mpegOutputTSConnection	mpegOutputTSNumPrograms	mpegOutputTSTSID	mpegOutputTSNitPid	n
1	100	100	mpts(2)	qam(2)	1.3.6.1.4.1.5591.1.11.5.4.2.6.1.100.1	2	1000	16	
2	137	137	mpts(2)	qam(2)	1.3.6.1.4.1.5591.1.11.5.4.2.6.1.137.1	2	1000	16	
3	138	138		qam(2)	1.3.6.1.4.1.5591.1.11.5.4.2.6.1.138.1	0	1038	16	

Figure I.1: Identifying a Replicated QAM by Looking at mpegOutputTSTSID

Annex J (informative): Vendor Schema Version in the CCAP XSD

The CCAP XSD provides a complex data type that allow the major, minor, and micro version numbers to be specified for a vendor-specific extension. This complex type may be used in a vendor-specific extension, but is not mandatory. The composition of this complex type is shown here:

```
<xs:complexType name="vendor-extension-version-type">
  <xs:sequence>
    <xs:element name="major-version" type="xs:unsignedInt" minOccurs="1" maxOccurs="1">
      <xs:annotation>
        <xs:documentation>
          Major version provides the macro versioning number for each interface.
          Versions containing the same major version should provide backwards
          compatibility.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="minor-version" type="xs:int" minOccurs="1" maxOccurs="1">
      <xs:annotation>
        <xs:documentation>
          MinorVersion identifies incremental and
          backwards compatible updates to a major version.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="micro-version" type="xs:int" minOccurs="0" maxOccurs="1">
      <xs:annotation>
        <xs:documentation>
          MicroVersion is usually for bug fixes, without changes in functionality.
        </xs:documentation>
       </xs:annotation>
    </xs:element>
    <xs:element name="ext" type="ext-type" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

Annex K (informative): Converting YANG to XSD

K.1 Using PYANG to Generate an XSD from the CCAP YANG Modules

The CCAP XML Schema is derived by an automated process that converts the standard CCAP YANG modules specified in Annex G into a valid schema file. The conversion of the YANG to XSD is performed by pyang-a YANG validator, transformer, and code generator, written in Python. While CableLabs manages the conversion of the CCAP YANG file into the CCAP schema file (XSD), pyang can be used by anyone to convert the CCAP YANG modules into a valid XSD.

Note that pyang requires Python to run; installing Python is beyond the scope of the present document.

To run pyang on the CCAP YANG module file to produce a valid instance of the CCAP XSD, complete the following steps:

1) Download the most recent version of pyang from the Google Code repository located at: http://code.google.com/p/pyang/downloads/

2) Install pyang according to the installation instructions found on this site.

3) Download the most recent version of the CCAP XSD translator plugin (ccapxsd.py) from the following location: http://www.cablelabs.com/YANG/DOCSIS/.

4) Place the ccapxsd.py file in the pyang/plugins directory; this directory will be located within the site local Python library directory.

5) Ensure that the following files are present in the local directory:

   - ietf-inet-types.yang (2010-09-24)
   - ietf-yang-types.yang (2010-09-24)
   - ccap@yyyy-mm-dd.yang (the most recent version of the CCAP YANG module file)

6) Run the pyang tool with the following command line options:

```
pyang -f ccapxsd --ccapxsd-global-complex-types --inline-type -o ccap@yyyy-mm-dd.xsd
ccap@yyyy-mm-dd.yang
```

   where yyyy-mm-dd represents the date on which the most recent version of the YANG module file was published.

This will produce an XML Schema file in the local directory.

It should be noted that pyang currently does not support creating a valid CCAP schema when vendor extensions to the standard CCAP YANG module file are included in a separate file.

K.2 Creating In-Line Data Types in the CCAP.XSD

During the conversion process, pyang converts containers and lists that have the same name to a complex type that can be reused throughout the model. While this conversion increases the extensibility of the configuration object model, there are cases in which these definitions should not be converted to a complex type; there is value in allowing them to be unique and extended on an individual basis.

To allow for these containers and lists to be extended on an individual basis, the YANG extension "inlineType" has been created. This extension, when placed in a container or list, inhibits the generation of a named complex type, leaving the type definition in-line. This allows the desired separate extension points for each occurrence.

One example of where this is useful is the ip-interface list in the virtual-interface-group. This grouping is included for both cable-bundle and loopback interfaces. Given that these interface types are very different, it is likely that a vendor would want to extend them differently, which is now allowed by the inLineType extension.

The following example shows this usage:

```
    grouping virtual-interface-group {
        leaf interface-index {
            type uint8;
            mandatory true;
            description "The index for this virtual ip-interface";
           }
        leaf admin-state {
            type admin-state-type;
            default down;
            description "This attribute configures the administrative state of the virtual
interface.";
        }
        list ip-interface {
            key ip-interface-name;
            max-elements 1;
            ccap:inlineType;
            description "An ip-interface object.";
            uses ip-interface-group;
            container yang-ext {
                ccap:extensionPoint; //different pyang flags impact use of this hint
                description "node for vendor YANG extensions";
            }
        }
    }
```

The extension is enabled on the pyang command line:

```
pyang -o ccap@2012-10-31.xsd -f ccapxsd --ccapxsd-global-complex-types
       --inline-type ccap@2012-10-31.yang
```

Annex L (informative): Bibliography

- CableLabs Assigned Names and Numbers, CL-SP-CANN-I09-130404, April 04, 2013, Cable Television Laboratories, Inc.
- CableLabs Definition MIB Specification, CL-SP-MIB-CLABDEF-I10-120809, August 9, 2012, Cable Television Laboratories, Inc.
- IETF RFC 3339, G. Klyne and C. Newman, Date and Time on the Internet: Timestamps, July 2002.
- IETF RFC 3411/STD0062, D. Harrington, et al., An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks, December 2002.

List of figures

List of Tables

History

Document history		
V1.1.1	November 2014	Membership Approval Procedure MV 20150123: 2014-11-24 to 2015-01-23