



***Society of Cable
Telecommunications
Engineers***

**ENGINEERING COMMITTEE
Data Standards Subcommittee**

AMERICAN NATIONAL STANDARD

ANSI/SCTE 137-5 2017

**Modular Headend Architecture Part 5: Edge QAM
Provisioning and Management Interface**

NOTICE

The Society of Cable Telecommunications Engineers (SCTE) Standards and Operational Practices (hereafter called “documents”) are intended to serve the public interest by providing specifications, test methods and procedures that promote uniformity of product, interchangeability, best practices and ultimately the long term reliability of broadband communications facilities. These documents shall not in any way preclude any member or non-member of SCTE from manufacturing or selling products not conforming to such documents, nor shall the existence of such standards preclude their voluntary use by those other than SCTE members.

SCTE assumes no obligations or liability whatsoever to any party who may adopt the documents. Such adopting party assumes all risks associated with adoption of these documents, and accepts full responsibility for any damage and/or claims arising from the adoption of such documents.

Attention is called to the possibility that implementation of this document may require the use of subject matter covered by patent rights. By publication of this document, no position is taken with respect to the existence or validity of any patent rights in connection therewith. If a patent holder has filed a statement of willingness to grant a license under these rights on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license, then details may be obtained from the standards developer. SCTE shall not be responsible for identifying patents for which a license may be required or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Patent holders who believe that they hold patents which are essential to the implementation of this document have been requested to provide information about those patents and any related licensing terms and conditions. Any such declarations made before or after publication of this document are available on the SCTE web site at <http://www.scte.org>.

All Rights Reserved

© Society of Cable Telecommunications Engineers, Inc. 2017
140 Philips Road
Exton, PA 19341

Note: DOCSIS® is a registered trademark of Cable Television Laboratories, Inc., and is used in this document with permission.

Contents

1	SCOPE	1
1.1	Introduction and Purpose	1
1.2	Requirements and Conventions	1
2	REFERENCES	2
2.1	Normative References	2
2.2	Informative References	3
2.3	Reference Acquisition	3
3	TERMS AND DEFINITIONS	4
4	ABBREVIATIONS AND ACRONYMS	5
5	THEORY OF OPERATION	6
5.1	Introduction	6
5.2	Provisioning	6
5.3	Configuration	6
6	EQAM PROVISIONING	7
6.1	Establishing IP Connectivity	7
6.1.1	EQAM Static Provisioning	8
6.1.2	EQAM Dynamic Provisioning	8
7	EQAM CONFIGURATION	11
7.1	Configuration File Download	11
7.2	Initial EQAM Configuration	13
7.3	EQAM Configuration with Administrative Interface	13
7.4	EQAM Configuration with XML File	14
7.5	Vendor-Specific Extensions	14
7.6	XML Configuration File Checksum	15
7.7	Configuration Error File	15
8	EQAM MANAGEMENT	16
8.1	DOCSIS MIB Modules	16
8.2	IETF RFC MIB Modules	16
8.3	SCTE MIB Modules	18
9	EQAM FAULT REPORTING	19
9.1	SNMP Usage	19
9.2	Event Notification	19
9.2.1	Format of Events	19
9.2.2	BIT Values for docsDevEvReporting [RFC 4639]	21
9.2.3	Standard events for EQAM	21
9.2.4	Event Priorities and Vendor-Specific Events	23
9.3	List of events	23
ANNEX A	DETAILED MIB REQUIREMENTS (NORMATIVE)	26
A.1	Object Model schema Definition	27
A.2	IF-MIB ifTable MIB-Object details	56
ANNEX B	DESCRIPTION OF THE XML BASED CONFIGURATION FILE	58

B.1	EqamCfg	60
B.1.1	System	61
B.1.2	Inputs	61
B.1.3	RFOutputs.....	61
B.1.4	Vendor Object.....	62
B.1.5	Checksum Object	63
B.2	Configuration File XML Schema Definition	63
B.2.1	EQAM-CFG_1.0.XSD.....	63
B.2.2	EQAM_1.0.XSD.....	64
B.2.3	DOCS-CABLE-DEVICE_RFC4639.XSD	69
B.2.4	SNMPv2_RFC3418.XSD	70
ANNEX C	DOCS-EQAM MANAGEMENT REQUIREMENTS	71
C.1	DOCS-EQAM Object Model Overview	71
C.2	DOCS-EQAM Object Model Definitions.....	71
C.2.1	DOCS-EQAM Object Model Data Types.....	71
C.2.2	DOCS-EQAM Object Model Class Diagram.....	72
C.2.3	DOCS-EQAM Object Model Description	73
C.3	EQAM MIB Module.....	89

Figures

FIGURE 6–1 - EQAM PROVISIONING FLOW	7
FIGURE 6–2 - IPV4 PROVISIONING MESSAGE FLOW.....	8
FIGURE 7–1 - EQAM CONFIGURATION PROCESS.....	12
FIGURE B–1 - XML CONFIGURATION OBJECT MODEL	59
FIGURE C–1 - QAM OBJECT MODEL DIAGRAM	72

Tables

TABLE 6–1 - DHCP BACKOFF DISTRIBUTION VALUES	9
TABLE 7–1 - EQAM CONFIGURATION WITH ADMINISTRATIVE INTERFACE	13
TABLE 7–2 - EQAM CONFIGURATION WITH XML FILE.....	14
TABLE 8–1 - DOCSIS MIB MODULES	16
TABLE 8–2 - IETF RFC MIB MODULES.....	16
TABLE 8–3 - SCTE MIB MODULES.....	18
TABLE 9–1 - DEFAULT EVENT REPORTING MECHANISM VERSUS PRIORITY (NON-VOLATILE LOCAL LOG SUPPORT ONLY)	22
TABLE 9–2 - DEFAULT EVENT REPORTING MECHANISM VERSUS PRIORITY (VOLATILE LOCAL LOG SUPPORT ONLY)	22
TABLE 9–3 - DEFAULT EVENT REPORTING MECHANISM VERSUS PRIORITY.....	23
TABLE 9–4 - EVENT PRIORITIES ASSIGNMENT	23
TABLE A–1 - MIB IMPLEMENTATION SUPPORT.....	26
TABLE A–2 - SNMP ACCESS REQUIREMENTS.....	26
TABLE A–3 - REQUIREMENTS.....	27
TABLE A–4 - IF-MIB IFTABLE MIB-OBJECT DETAILS	56
TABLE B–1 - EQAMCFG CONFIG FILE ELEMENTS.....	60
TABLE B–2 - SYSTEM OBJECT.....	61
TABLE B–3 - INPUTS ELEMENTS	61
TABLE B–4 - EDGEINPUTS ELEMENTS	61
TABLE B–5 - RFOUTPUTS ELEMENTS.....	61
TABLE B–6 - RFPORTS ELEMENTS	62
TABLE B–7 - RFPORT ELEMENTS	62
TABLE B–8 - QAMCHANNELS ELEMENTS	62
TABLE B–9 - CHANNEL ELEMENTS.....	62
TABLE B–10 - VENDOR OBJECT.....	63
TABLE B–11 - CHECKSUM OBJECT.....	63
TABLE C–1 - CONTROL OBJECT	73
TABLE C–2 - NMSACCESS OBJECT.....	74
TABLE C–3 - TIME OBJECT	75
TABLE C–4 - SYSLOGSERVER OBJECT	76
TABLE C–5 - REGISTRATION OBJECT	76
TABLE C–6 - TSINPUTCFG OBJECT.....	78
TABLE C–7 - TSOUTCFG OBJECT	79
TABLE C–8 - RESERVEDUDPMP OBJECT	79
TABLE C–9 - RESERVEDPIDRANGE OBJECT.....	80
TABLE C–10 - DEPICFG OBJECT	80
TABLE C–11 - EDGEINPUT OBJECT	80
TABLE C–12 - INPUTREGISTRATION.....	81
TABLE C–13 - RFPORT OBJECT	81
TABLE C–14 - FIBERNODE OBJECT.....	83
TABLE C–15 - CHANNEL OBJECT	83
TABLE C–16 - TSOUTCFG OBJECT	85
TABLE C–17 - STATICUDPMP OBJECT.....	86
TABLE C–18 - SESSIONLOGCTRL	86
TABLE C–19 - SESSIONLOG.....	87
TABLE C–20 - NOTIFY OBJECT.....	89

This page left blank intentionally.

1 SCOPE

NOTE: This document is identical to SCTE 137-5 2010 except for informative components which may have been updated such as the title page, NOTICE text, headers and footers. No normative changes have been made to this document.

1.1 Introduction and Purpose

This specification is a component of the Modular Headend Architecture; in particular it defines the Provisioning and Management requirements for the EQAM device.

1.2 Requirements and Conventions

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

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

2 REFERENCES

2.1 Normative References

The following documents contain provisions, which, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and users of this document are encouraged to investigate the possibility of applying the most recent editions of the documents listed below.

- [CANN DHCP] CableLabs DHCP Options Registry Specification, CL-SP-CANN-DHCP-Reg-I03-090811, August 11, 2009, Cable Television Laboratories, Inc.
- [PW-MIB] IETF draft-ietf-pwe3-pw-mib-14.txt, Zelig, D., Nadeau, T., Pseudo Wire (PW) Management Information Base, January 2008, <http://tools.ietf.org/html/draft-ietf-pwe3-pw-mib-14>.
- [RFC 1123] IETF RFC 1123/STD0003, R. Braden, Editor, Requirements for Internet Hosts -- Application and Support, October 1989.
- [RFC 1350] IETF RFC 1350/STD0033, K. Sollins, The TFTP Protocol (Revision 2), July 1992.
- [RFC 2011] IETF RFC 2011, Category: Standards Track SNMPv2 Management Information Base for the Internet Protocol using SMIPv2, November 1996.
- [RFC 2131] IETF RFC 2131, R. Droms, Dynamic Host Configuration Protocol, March 1997.
- [RFC 2326] IETF RFC 2326, H. Schulzrinne, Columbia U., R. Lanphier, Real Time Streaming Protocol (RTSP), April 1998.
- [RFC 2348] IETF RFC 2348, G. Malkin, A. Harkin, TFTP Blocksize Option, May 1998.
- [RFC 2661] IETF RFC 2661, Layer Two Tunneling Protocol "L2TP", August 1999.
- [RFC 2863] IETF RFC 2853, The Interfaces Group MIB, June 2000.
- [RFC 3164] IETF RFC 3164, C. Lonvick, The BSD syslog Protocol, August 2001.
- [RFC 3203] IETF RFC 3203, Y. T'Joens, C. Hublet, P. DeSchrijver, DHCP reconfigure extension, December 2001.
- [RFC 3371] IETF RFC 3371, L2TPv3 Extensions Working Group, Layer Two Tunneling Protocol 'L2TP' Management Information Base, August 2002.
- [RFC 3411] IETF RFC 3411/STD0062, An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks, December 2002.
- [RFC 3412] IETF RFC 341 /STD0062, Message Processing and Dispatching for the Simple Network Management Protocol (SNMP), December 2002.
- [RFC 3413] IETF RFC 3413/STD0062, Simple Network Management Protocol (SNMP) Applications, December 2002.
- [RFC 3414] IETF RFC 3414/STD0062, User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3), December 2002.
- [RFC 3415] IETF RFC 3415/STD0062, View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP), December 2002.
- [RFC 3418] IETF RFC 3418/STD0062, Management Information Base (MIB) for the Simple Network Management Protocol (SNMP), December 2002.
- [RFC 3584] IETF RFC 3584, Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework, August 2003.
- [RFC 4087] IETF RFC 4087, D. Thaler, IP Tunnel MIB, June 2005.
- [RFC 4133] IETF RFC 4133, A. Bierman, K. and McCloghrie, Entity MIB, August 2005.
- [RFC 4291] IETF RFC 4291, R. Hinden, S. Deering, IP Version 6 Addressing Architecture, February 2006.

- [RFC 4293] IETF RFC 4293, S. Routhier, Ed., Management Information Base for the Internet Protocol (IP), April 2006.
- [RFC 4361] IETF RFC 4361, T. Lemon and B. Sommerfield, Node-specific Client Identifiers for Dynamic Host Configuration Protocol Version Four (DHCPv4), February 2006.
- [RFC 4546] IETF RFC 4546, D. Raftus and E. Cardona, Radio Frequency (RF) Interface Management Information Base for DOCSIS 2.0 Compliant RF Interfaces, June 2006.
- [RFC 4639] IETF RFC 4639, R. Woundy and K. Marez, Cable Device Management Information Base for Data-Over-Cable Service Interface Specification (DOCSIS) Compliant Cable Modems and Cable Modem Termination Systems, December 2006.
- [SCTE 137-3] ANSI/SCTE 137-3 2007, Modular Headend Architecture Part 3: Operations Support System Interface. (Formerly known as SCTE 141.)
- [SCTE 137-4] ANSI/SCTE 137-4 2007, Modular Headend Architecture Part 4: Edge Resource Manager Interface. (Formerly known as SCTE 139.)
- [SCTE 154-2] ANSI/SCTE 154-2 2008, SCTE-HMS-QAM-MIB Management Information Base (MIB) Definition.
- [SCTE 154-4] ANSI/SCTE 154-4 2008, MPEG Management Information Base.
- [SHA-1] Federal Information Processing Standards Publication 180-2, 2002 August 1, Announcing the Secure Hash Standard.

2.2 Informative References

- [SCTE 137-7] SCTE 137-7 2010, Modular Headend Architecture Part 7: EQAM Architectural Overview Technical Report

2.3 Reference Acquisition

- Cable Television Laboratories, Inc., 858 Coal Creek Circle, Louisville, CO 80027; Phone +1-303-661-9100; Fax +1-303-661-9199; <http://www.cablelabs.com>
- International Telecommunication Union – Telecommunication Standardization Sector (ITU-T), <http://www.itu.int/itu-t/>
- Federal Information Processing Standards (FIPS-PUBS); <http://www.itl.nist.gov/fipspubs/by-num.htm>
- Society of Cable Telecommunication Engineers, 140 Philips Road, Exton PA 19341-1318, Phone +1-610-524-1725; Internet: <http://www.scte.org> / E-Mail: standards@scte.org

3 TERMS AND DEFINITIONS

This specification uses the following terms:

Edge QAM	A head-end or hub device that receives packets of digital video or data from the operator network. It re-packetizes the video or data into an MPEG transport stream and digitally modulates the transport stream onto a downstream RF carrier using QAM.
MAC domain	A grouping of layer 2 devices that can communicate with each other without using bridging or routing. In DOCSIS, a MAC domain is the group of CMs that are using upstream and downstream channels linked together through a MAC forwarding entity.
Service Group	An HFC service group (also known as a service group) is a portion of an HFC access network used to deliver a set of services to a population of cable modems that share a common spectrum of RF channels.

4 ABBREVIATIONS AND ACRONYMS

This specification uses the following abbreviations:

EQAM	Edge QAM
ERM	Edge Resource Manager
HFC	Hybrid Fiber/Coaxial Cable
IF	Interface
IEEE	Institute of Electrical and Electronics Engineers
IGMP	Internet Group Management Protocol
IP	Internet Protocol
MPEG	Moving Pictures Experts Group
MPTS	Multiple Program Transport Stream
NMS	Network Management System
QAM	Quadrature Amplitude Modulator
RF	Radio Frequency
RFC	Request for Comment
SDV	Switched Digital Video
SPTS	Single Program Transport Stream
SSM	Source Specific Multicast
XML	Extensible Markup Language
XSID	XML Schema Definition

5 THEORY OF OPERATION

The role of the EQAM in the video-on-demand and switched-digital-video architecture is to receive an IP unicast or multicast stream containing MPEG transport stream packets, and then produce that transport stream on one or more RF outputs for transmission over the hybrid fiber-coax cable plant.

5.1 Introduction

This specification covers the configuration, provisioning, and management requirements for an EQAM device. The Provisioning section provides details on how the EQAM is initially provisioned manually, as well as via DHCP-based provisioning. The Configuration section provides details on how the configuration management settings are applied and saved at the EQAM. The Fault Management section provides details on error-handling requirements for the EQAM. Finally, the Management requirements are covered in the Annex sections of this specification.

5.2 Provisioning

The EQAM management interface needs to be provisioned with an IP address and other configuration parameters. The EQAM IPv4 provisioning is specified to allow Operators flexibility with either static or dynamic using DHCPv4 provisioning process.

5.3 Configuration

The XML configuration file is used to set the EQAM parameters value. The configuration file must always be fully reapplied at boot up time. At any time, the EQAM must be able to apply the common section of the XML configuration described in this specification. The configuration file can be saved locally on the EQAM (with non-volatile memory) or remotely on a centralized file server.

When the configuration files are on a server, they can be downloaded to the EQAM with TFTP or HTTPS via manual or DHCP control. The file can also be uploaded to the server to keep an updated copy when the EQAM configuration is modified by other administrative interfaces (CLI/GUI/SNMP). To keep some flexibility, specific vendor extensions can be included in a dedicated section of the file.

6 EQAM PROVISIONING

6.1 Establishing IP Connectivity

The EQAM's Management interface requires an IP address and other parameters before it can download its initial configuration file. Static as well as dynamic (DHCP) provisioning will be available to provision the IP parameters for the EQAM Management interface. The EQAM provisioning flow is outlined in Figure 6–1.

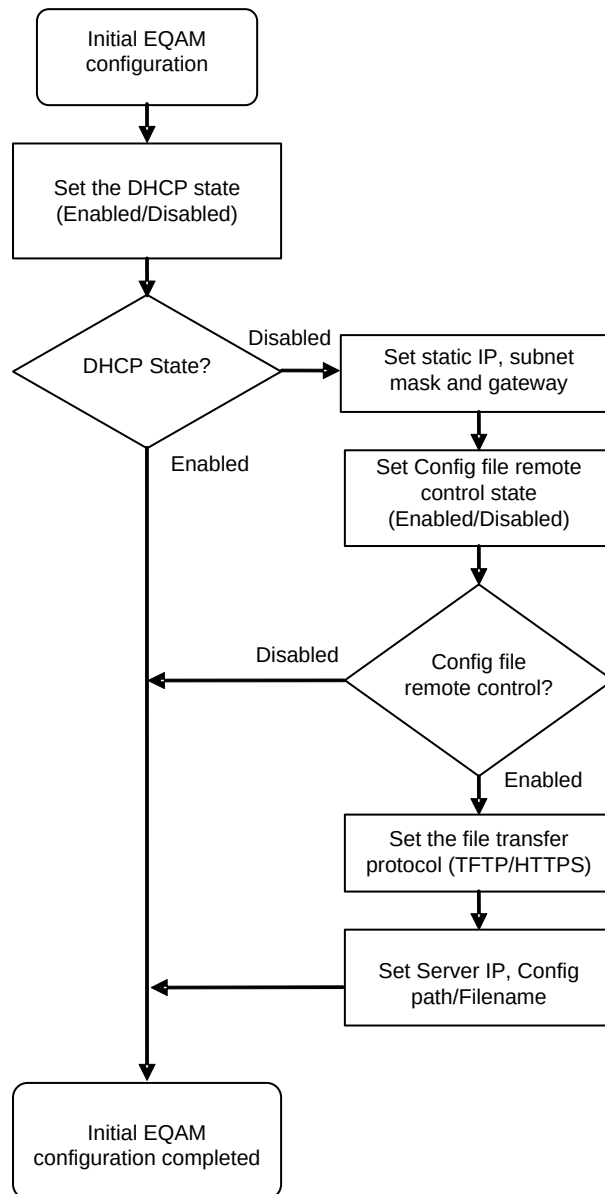


Figure 6–1 - EQAM Provisioning Flow

6.1.1 EQAM Static Provisioning

The EQAM MUST provide a user interface for provisioning the IP address, the IP mask, the default router address, the TFTP server address, and the configuration file name for its Management interface.

When shipped, the EQAM MUST be able to perform a standalone boot with the following factory configuration:

- Known static IP address (192.168.0.1)
- RF ports disabled

6.1.2 EQAM Dynamic Provisioning

The EQAM MUST allow the selection of DHCP for provisioning of its Management Interface.

6.1.2.1 Establish IPv4 Network Connectivity using DHCPv4

This section describes how the EQAM is dynamically provisioned with an IPv4 address and associated parameters.

The standard DHCPv4 message exchange between the EQAM and the DHCPv4 server is shown in Figure 6–2. The EQAM MUST use DHCPv4 [RFC 2131] in order to obtain an IP address and other parameters needed to establish IP connectivity.

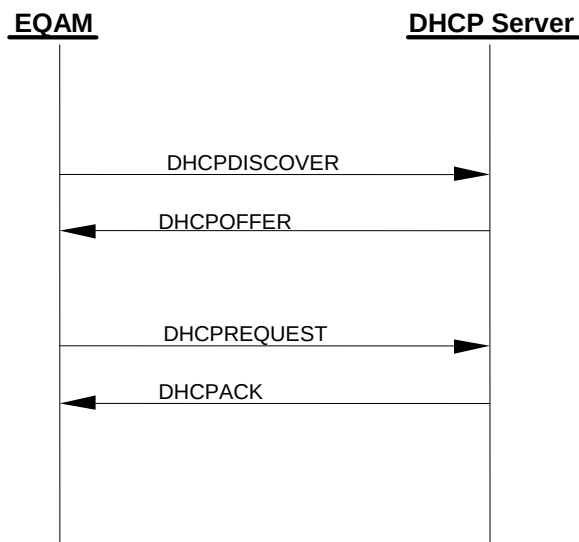


Figure 6–2 - IPv4 Provisioning Message Flow

The EQAM may receive multiple DHCPOFFER messages in response to its DHCPDISCOVER message. If a received DHCPOFFER message does not include all of the required DHCPv4 fields and options as described in Section 6.1.2.1.1, the EQAM MUST discard the DHCPOFFER message and wait for another DHCPOFFER message. If none of the received DHCPOFFER messages contain all the required DHCPv4 fields and options, the EQAM retransmits the DHCPDISCOVER message.

The EQAM SHOULD choose backoff values for retransmission of DHCPDISCOVER messages according to a uniform distribution between the minimum and maximum values in the rows of the table below.

Table 6–1 - DHCP Backoff Distribution Values

Backoff Number	Minimum (sec.)	Maximum (sec.)
1	3	5
2	7	9
3	15	17
4	31	33
5	63	65

The EQAM SHOULD also implement a different retransmission strategy for the RENEWING and REBINDING states as recommended in [RFC 2131], which is based on one-half of the remaining lease time.

The EQAM MUST limit the number of retransmissions to five or fewer for the DHCPDISCOVER and DHCPREQUEST messages.

[RFC 3203] describes an extension to DHCPv4 that allows a DHCP server to send a FORCERENEW message that forces a client to renew its lease. The EQAM MUST ignore all received FORCERENEW messages.

6.1.2.1.1 DHCPv4 Fields Used by the EQAM

The following fields MUST be present in the DHCPDISCOVER and DHCPREQUEST messages from the EQAM:

- The hardware type (htype) MUST be set to 1;
- The hardware length (hlen) MUST be set to 6;
- The client hardware address (chaddr) MUST be set to the 48-bit MAC address associated with the Management interface of the EQAM;
- The client identifier option MUST be included using the format defined in [RFC 4361];
- The parameter request list option MUST be included. The following option codes defined in [RFC 2131] MUST be included in the list:
 - Option code 1 (Subnet Mask)
 - Option code 2 (Time Offset)
 - Option code 3 (Router Option)
 - Option code 4 (Time Server Option)
 - Option code 7 (Log Server Option)
- Option code 60 (Vendor Class Identifier) - the following ASCII-encoded string MUST be present in Option code 60":EQAM:"
- Option code 125 (DHCPv4 Vendor - Identifying Vendor Specific Information Options for DOCSIS 3.0 defined in [CANN DHCP] includes the following sub-options:
 - Sub-option code 1, the DHCPv4 Option Request option. This option code MUST be included in the DHCPv4 Option Request option.

- Sub-option code 2, DHCPv4 TFTP Servers Option.

The following fields are expected in the DHCP OFFER and DHCP ACK messages returned to the EQAM. The EQAM MUST configure itself with the listed fields from the DHCP ACK:

- The IP address to be used by the EQAM (yiaddr) (critical);
- The IP addresses of the TFTP servers for use in the next phase of the boot process (DHCPv4 TFTP Servers option defined in [CANN DHCP]) (optional);
- The name of the EQAM configuration file to be read from the TFTP server by the EQAM (file) (optional);
- The subnet mask to be used by the EQAM (Subnet Mask, option 1) (non-critical);
- The time offset of the EQAM from UTC (Time Offset, option 2). This is used by the EQAM to calculate a time for use in error logs (non-critical).
- A list of addresses of one or more routers to be used for forwarding IP traffic originating from the EQAM's IP stack (Router Option, option 3). The EQAM is not required to use more than one router IP address for forwarding (non-critical).
- A list of ToD servers from which the current time may be obtained (Time Server Option, option 4) (non-critical).
- A list of syslog servers to which logging information may be sent (Log Server Option, option 7) (non-critical).

If a critical field is missing or invalid in the DHCP ACK received during initialization, the EQAM MUST:

1. Log an error.
2. Proceed as if the acquisition of the IPv4 address through DHCPv4 has failed.

If a non-critical field is missing or invalid in the DHCP ACK received during initialization, the EQAM MUST log a warning, ignore the field, and continue the IPv4 provisioning process.

If the yiaddr field is missing or invalid in the DHCP ACK received during a renew or rebind operation, the EQAM MUST log an error and restart DHCPv4 provisioning.

If any other critical or non-critical field is missing or invalid in the DHCP ACK received during a renew or rebind operation, the EQAM MUST log a warning, ignore the field if it is invalid, and remain operational.

6.1.2.1.2 Use of T1 and T2 Timers

The EQAM MUST initiate the lease renewal process when timer DHCP-T1 expires. The EQAM MUST initiate the lease rebinding process when timer DHCP-T2 expires. Timers DHCP-T1 and DHCP-T2 are called T1 and T2, respectively, in the DHCP specifications. If the DHCP server sends a value for DHCP-T1 to the EQAM in a DHCP message option, the EQAM MUST use that value. If the DHCP server does not send a value for DHCP-T1, the EQAM MUST set DHCP-T1 to one-half of the duration of the lease [RFC 2131]. If the DHCP server sends a value for DHCP-T2 to the EQAM in a DHCP message option, the EQAM MUST use that value. If the DHCP server does not send a value for DHCP-T2, the EQAM MUST set DHCP-T2 to seven-eighths of the duration of the [RFC 2131].

6.1.2.1.2.1 DHCPv4 Renew Fields Used by the EQAM

It is possible during the DHCPv4 renew operation that the EQAM will receive updated fields in the DHCP ACK message. If the EQAM is unable to use the updated values of those fields, it may discard the renew operation and restart DHCP.

7 EQAM CONFIGURATION

The EQAM either uses a locally stored configuration or tries to download a configuration file using TFTP from the File Server.

If the EQAM is statically provisioned with its IP Address information and the configuration file download information is also supplied, then it uses the configuration file download mechanism via TFTP as specified below. If the configuration file download information is left out during static provisioning, then the EQAM uses the locally stored configuration.

If the EQAM is dynamically provisioned via DHCPv4 for its IP Address information and the configuration file download information is also supplied through DHCPv4, then it uses the configuration file download mechanism via TFTP as specified below. If the configuration file download information is left out during DHCPv4 provisioning, then the EQAM uses the locally stored configuration.

Figure 7–1 shows the configuration process for the EQAM. The next section defines how an EQAM downloads its configuration file and uses the configuration parameters to initialize its resources.

7.1 Configuration File Download

The EQAM **MUST** use TFTP to obtain a configuration file from the server(s) identified in the DHCPv4 TFTP Servers option, when provided. The EQAM **MUST** include the TFTP Blocksize option [RFC 2348] when requesting the configuration file. The EQAM **MUST** request a blocksize of 1428 if using TFTP over IPv4.

There may be more than one value in the DHCPv4 TFTP Servers option in the DHCPACK; the EQAM **MUST** use the values in this option to obtain its configuration file.

The EQAM **MUST** use the following procedure to attempt configuration file download:

- Initiate a configuration file download by sending a TFTP read request message for the configuration file to the TFTP server, thus establishing a connection with the server [RFC 1350].
- If the EQAM is unable to download the configuration file on the first try successfully, the EQAM **MUST** retry the TFTP read request as defined below.
- The EQAM **MUST** use an adaptive timeout based on binary exponential backoff until the EQAM successfully receives the requested file from the server as follows:
 1. Each retry is 2^n second(s) following the previous request, where the EQAM configuration file retry counter, $n = \{0, 1, 2 \dots \text{MaxRetryCounter}-1\}$. The default MaxRetryCounter is 5.
 2. The retry counter $n = 0$ for the first retry, then is incremented by one for each subsequent request until $n = (\text{MaxRetryCounter}-1)$.
- If the EQAM does not successfully acquire the requested configuration file following the attempt with $n = (\text{MaxRetryCounter}-1)$, n is to be reset to 0 and the EQAM moves to next Server IP if present and repeats the above procedure.
- Once the EQAM reaches the end of the DHCPv4 TFTP Servers List, it increments the TFTP retry attempts count by one and repeats the above procedure for each Server IP until TFTP retry attempts count equals MaxRetryAttempt. By default, MaxRetryAttempt must be 3.

If WaitConfigFile value is FALSE (2) after completing the above procedure without successfully downloading a configuration file, the EQAM MUST utilize the locally stored configuration. By default, WaitConfigFile MUST be FALSE(2).

If WaitConfigFile is set to TRUE (1) after completing the above procedure without successfully downloading a configuration file, the EQAM MUST re-start the DHCP process from the beginning.

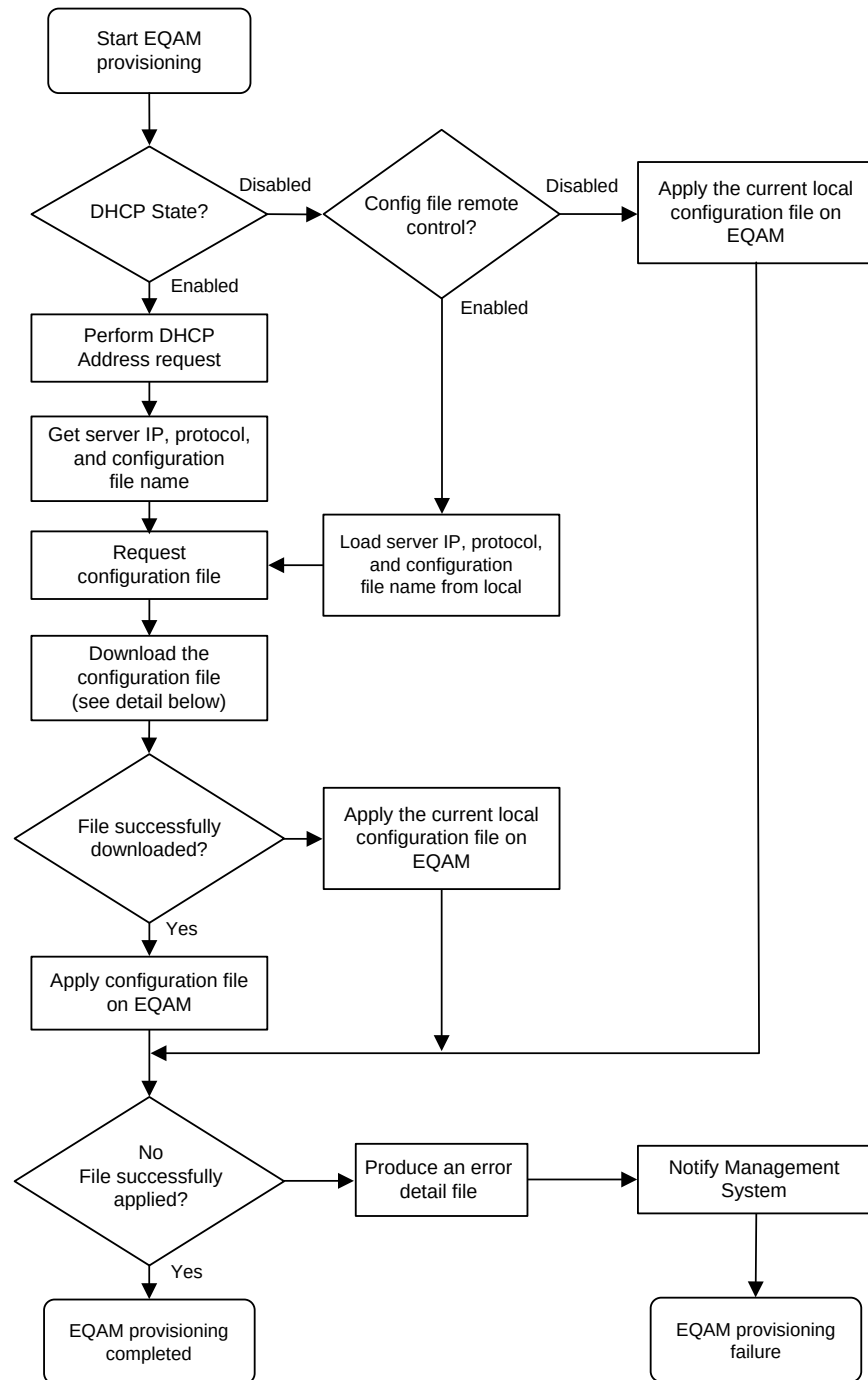


Figure 7-1 - EQAM Configuration Process

7.2 Initial EQAM Configuration

Once the EQAM IP configuration is done and the configuration file remote control is activated, an initial configuration file can be used to set up the EQAM with a known default state.

The initial configuration file contains some basic default configuration values for the individual headend/serving groups (based on the fact that not all headends/serving groups are configured the same) and different types of EQAM equipment (vendor differences, as well as model differences).

7.3 EQAM Configuration with Administrative Interface

Once the EQAM is operational, configuration settings on the EQAM can be modified via an administrative interface. The administrative interface includes a small set of standard SNMP RW objects as described in Annex A, as well as any proprietary interfaces developed by the vendor, e.g., proprietary SNMP MIBs, CLI, WEB interface, etc.

As an example, as part of the staging of a new EQAM within an MSO network, specific information for the EQAM is configured, such as QAM frequencies, TSID values, TSID groups, Serving Group Names, QAM Group Names, DEPI session IDs, etc.

There **MUST** be a configuration option at the EQAM, via an Administrative Interface, to allow the EQAM to store the updated configuration in local non-volatile storage.

There **MUST** be a configuration option at the EQAM, via an Administrative Interface, to allow the EQAM to generate an XML configuration file and upload it via TFTP to a remote configuration file server with a specific file name.

Table 7–1 summarizes the steps involved in modifying the EQAM configuration and uploading it using TFTP.

Table 7–1 - EQAM Configuration with Administrative Interface

	Management System	EQAM	Config Server	Description
1				Administrative interfaces (CLI/SNMP/WEB) used to modify EQAM configuration.
2				EQAM notifies MGT system about reconfiguration.
3				Administrative interface triggers generation and upload of XML file.
4				XML file transfer from EQAM to server to ensure up to date copy.

Given the fact that the EQAM-created XML configuration file contains the most up-to-date configuration information for this particular EQAM, the operator can use this interface to maintain a copy of the configuration details of the EQAM remotely. The same XML file can be used to replace the configuration file pointed to by DHCP (if the EQAM uses DHCP).

7.4 EQAM Configuration with XML File

Often, the vendor may need to make wide scale changes for large numbers of EQAMs in a certain serving group or region. Using the supplied specific interfaces on EQAMs to reconfigure them may prove to be unwieldy in the case of large numbers of EQAMs. Having the ability to modify the specified XML details within the per-EQAM configuration files, and subsequently forcing a re-download/application of the EQAM configuration files to the EQAMs in question, simplifies the configuration change process.

The EQAM MUST support a configuration option via the Administrative Interface in order to trigger the XML configuration file to be downloaded from a remote configuration file server (defined by a specific FILENAME), and subsequently applied to the EQAM configuration. The EQAM may or may not take into account any vendor-specific extensions contained in the configuration file during this update process.

When the downloaded configuration file contains updates to the QAM channel parameter configuration, the EQAM 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.

Table 7–2 resumes the EQAM configuration step with the XML file.

Table 7–2 - EQAM Configuration With XML File

	Management System	EQAM	Config Server	Description
1				XML configuration file(s) modified by operator on the file server.
2				EQAM administrative interface sets up file transfer detail.
3				EQAM administrative interface triggers download of XML file.
4				XML configuration file is applied on EQAM.
5				EQAM notifies MGT if the configuration file can not be applied correctly.

7.5 Vendor-Specific Extensions

The vendor-specific extensions are a group in the XML file under the "VendorConfig" element. This element MUST include a "vendorId" attribute that uses the value of the private enterprise number (PEN) assigned by IANA. Other specific vendor extensions may be included in the "VendorConfig" element with XML special case "CDATA" construct or with standard XML sub elements. Those extensions are proprietary to the vendor. (The proprietary content is not defined here.)

The vendor configuration is applied first, followed by the standard XML configuration details. For example, if the vendor-specific configuration information contains a frequency of 800 MHz for a particular Downstream QAM channel; the content of the XML configuration would override the vendor-specific configuration, thus ensuring that the standard XML configuration options are always applied. This also applies in the case where an EQAM has a local configuration file, not part of the XML configuration file. The XML configuration file is always applied AFTER the local configuration file is applied.

7.6 XML Configuration File Checksum

The XML configuration file is downloaded and uploaded via TFTP between the EQAM and the file server. There exists a possibility that contents of the XML configuration file may get corrupted or lost during a TFTP transfer.

In order to prevent this from affecting EQAM operation, a simple procedure is defined here that employs the use of a checksum inserted as an XML tag in the XML configuration file.

The EQAM scans the XML configuration file after any TFTP download and if the XML tag for the checksum is present; then the EQAM verifies that the contents of the XML configuration file correspond to the checksum contained within. A [SHA-1] hash-based checksum is specified for use in the XML tag.

If the XML tag containing the checksum is present in the XML configuration file downloaded at the EQAM, the EQAM MUST remove the XML checksum tag and validate the contents of the file by re-calculating the [SHA-1] hash-based checksum over the remaining contents of the XML file. If the checksum validation fails, the EQAM MUST generate an appropriate event and reject the configuration file that it has downloaded.

The EQAM MUST provide a configuration mechanism by which the operator may invoke the EQAM to generate and include the XML checksum tag every time the EQAM generates the XML configuration file. The default value of this configuration is FALSE(2).

The schema for the XML checksum tag is defined in Annex B.2.

7.7 Configuration Error File

The EQAM SHOULD validate all the parameters supplied in the configuration file before applying any updates. If error(s) are encountered during configuration, then it MUST NOT apply any of the configuration settings from the file. If the configuration file cannot be applied on the EQAM, the EQAM MUST continue to operate with the configuration it had in place prior to the download.

When the configuration file cannot be applied on the EQAM, the EQAM MUST log an event (EventId 079XXXXX). The error must be in the following comma-separated field format.

Line number, Element/Attribute name, Error description

Example:

123, qamChannelFrequency, wrong value
456, qamId, unknown QAM channel
569, tsOutput, syntax error (missing ">")

The operator will get meaningful debugging information if carriage returns are applied as appropriate in the configuration file.

8 EQAM MANAGEMENT

This section defines the minimum set of managed objects required to be supported by the EQAM.

The requirements described in this specification have priority over other standard organizations' defined MIB modules (e.g., SCTE, IETF). They include MIB objects made mandatory in this specification, whereas the other standards may have defined them as deprecated, obsolete, or as having optional implementation compliances.

Unless otherwise indicated, the other standard organizations' defined deprecated, obsolete, or optional MIB objects that are supported by the EQAM device **MUST** be implemented correctly according to the MIB module definition.

If an EQAM device does not support a deprecated, obsolete, or optional MIB object, the device SNMP Agent **MUST NOT** instantiate the MIB object. If an EQAM device does not support a deprecated, obsolete, or optional MIB object, the device SNMP Agent **MUST** return the corresponding error code on SNMP PDU requests.

8.1 DOCSIS MIB Modules

Table 8–1 - DOCSIS MIB Modules

MIB	OSSI Requirements
DOCSIS M-CMTS Interface MIB: DOCS-IF-M-CMTS-MIB	A compliant EQAM device MUST support the DOCS-IF-M-CMTS-MIB module as described in Annex A.
DOCSIS Time Interface MIB: DTI-MIB	A compliant EQAM device MUST support the DTI-MIB module as described in Annex A.
DOCSIS QAM MIB: DOCS-QAM-MIB	A compliant EQAM device MUST support the DOCS-QAM-MIB module as described in Annex C.
DOCSIS DRF MIB: DOCS-DRF-MIB	A compliant EQAM device MUST support the DOCS-DRF-MIB module as described in Annex A.

8.2 IETF RFC MIB Modules

Table 8–2 - IETF RFC MIB Modules

MIB	OSSI Requirements
SNMPv2 Management Information Base for the Internet Protocol using SMIPv2: IP-MIB	A compliant EQAM device MUST support ipNetToMediaTable and ipAddrTable from the ipGroup [RFC 2011]. The EQAM-compliant device MAY support other objects of the ipGroup.
The Interfaces Group MIB using SMIPv2: IF-MIB	The Interface MIB [RFC 2863] MUST be implemented by compliant M-CMTS Cores, DTI Server, and EQAM devices, as described in Annex A. A compliant EQAM device MUST support the interface Type docsCableMCmtsDownstream for each QAM channel. A compliant EQAM device MUST have an instance of ifEntry for each M-CMTS Downstream Interface. The EQAM-compliant device MUST include entries in the ifStackTable [RFC 2863] of the M-CMTS Downstream interfaces. A compliant EQAM device MUST model the docsCableMCmtsDownstream

MIB	OSSI Requirements
	interfaces at the top of the interface stack with no sub-interfaces.
Layer Two Tunneling Protocol "L2TP" Management Information Base [RFC 3371], August 2002.	The DOCS-IF-M-CMTS-MIB provides mechanisms for static configuration of DEPI L2TPv3 tunnels, as well as providing status information on dynamic DEPI L2TPv3 sessions created by other means, such as [SCTE 137-4]. The MIB table docsIfMCmtsDepsessionConfigTable follows a similar structure of Pseudo Wire (PW) MIB [PW-MIB]. Therefore, the current L2TP-MIB [RFC 3371] reference to the TUNNEL-MIB [RFC 4087] is no longer needed. The current L2TP-MIB is based on L2TP protocol [RFC 2661] and has not been updated for L2TPv3 Pseudowire framework. As a result, some information and capability developed for L2TPv3 has not been reflected in the MIB. Because of this, the use of the L2TP-MIB [RFC 3371] is not required for a compliant M-CMTS Core or EQAM.
SNMP v3 MIBs: • SNMP-FRAMEWORK-MIB, • SNMP-MPD-MIB, • SNMP-NOTIFICATION-MIB, • SNMP-TARGET-MIB, • SNMP-USER-BASED-SM-MIB, • SNMP-VIEW-BASED-ACM-MIB, • SNMP-COMMUNITY-MIB	A compliant EQAM device MUST support the SNMP v3 MIBs, as described in Annex A.
Management Information Base (MIB) for the Simple Network Management Protocol (SNMP): SNMPv2-MIB	The EQAM-compliant device MUST support the systemGroup of [RFC 3418]. Compliant EQAM and DTI Servers MUST support the MIB object snmpEnableAuthenTraps from the snmpGroup, and the notifications coldStart and authenticationFailure from the snmpBasicNotificationsGroup of [RFC 3418].
Entity MIB: ENTITY-MIB	A Compliant EQAM device MUST support the ENTITY-MIB objects as described in Annex A and [RFC 4133]. Compliant QAM and DTI devices are not required to implement Logical Management Entities, as defined in [RFC 4133] (e.g., to manage multiple EQAM chassis as logical entities). Therefore, support of MIB objects from entLogical2Group is not required. A Compliant EQAM device MAY implement Logical Management Entities, in which case the entLogical2Group MUST be supported. A compliant EQAM MUST assign the ENTITY-MIB PhysicalClass type of 'port' to QAM channels. A compliant EQAM device MUST assign RF Ports the PhysicalClass 'module'. Other Physical Class types, such as 'stack,' 'chassis,' 'backplane,' 'module,' etc., are used as the ENTITY-MIB [RFC 4133] describes them. A compliant EQAM device MUST map Ethernet interfaces and QAM channels (Physical entities of Entity PhysicalClass equals 'port') to ifEntry interfaces by means of entAliasMappingEntry with entAliasMappingIdentifier equal to the

MIB	OSSI Requirements
	respective instance of ifIndex.
DOCSIS RFI MIB: DOCS-IF-MIB	The EQAM device MUST implement the conceptual table docsIfDownstreamTable as indicated in Annex A. The EQAM MUST report the transmit power in the MIB object docsIfDownChannelPower within 2 dB of the actual transmit power.
DOCSIS Cable Device MIB: DOCS-CABLE-DEVICE-MIB	The EQAM device MUST implement the docsDevEventGroup from [RFC 4639] as indicated in Annex A.
IP-MIB	The EQAM device MUST implement the IP-MIB [RFC 4293] as indicated in Annex A. The EQAM device MUST instantiate a row in the ipIfStatsTable for each Management interface. The EQAM device MAY instantiate a row in the ipIfStatsTable for each Data interface. The EQAM device MUST implement the counters in the IpGroup to reflect Management interface statistics. The EQAM device MAY include the Data interface statistics in the IpGroup counters.

8.3 SCTE MIB Modules

Table 8-3 - SCTE MIB Modules

MIB	OSSI Requirements
Appendix I of SCTE-HMS-QAM-MIB	The EQAM device MUST implement the SCTE-HMS-QAM-MIB from [SCTE 154-2] as defined in Annex A.
Appendix III of SCTE-HMS-MPEG-MIB	The EQAM device MUST implement the SCTE-HMS-MPEG-MIB from [SCTE 154-4] as defined in Annex A.
A draft extension to the SCTE-HMS-MPEG-MIB [SCTE 154-4]	Input TS data rate would be added as an extension table. inputTSStatsExtTable Data rate: data rate for the entire input transport stream. Bits per second. Input program stats table Data rate: data rate associated with each individual program. Bits per second. Output program stats table Data rate: data rate associated with each individual program. Bits per second. Dropped packets: Count for dropped packets for each individual program. (Note: Should be same syntax as OutputStatsDroppedPackets, count 32, need to verify). Re-introduce the Video Session Log table. This is optional as per the earlier FT discussions.

9 EQAM FAULT REPORTING

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

9.1 SNMP Usage

SNMP is used to achieve the goals of fault management: remote detection, diagnosis, reporting, and correction of EQAM faults. The EQAM **MUST** support SNMP management traffic across the Management interface.

The EQAM SNMP access might be restricted by configuration parameters to support the operator's policy goals. Cable operators can use SNMP queries to perform diagnostics and fault classification.

The EQAM sends SNMP notifications to one or more NMSs (subject to operator-imposed policy). The EQAM sends events to a syslog server.

9.2 Event Notification

The EQAM is required to generate asynchronous events that indicate malfunction situations and notify about important events. The methods for reporting events are defined below:

1. Stored in Local Log (docsDevEventTable [RFC 4639]).
2. Reported to SNMP entities as an SNMP notification.
3. Sent as a message to a syslog server.

Event Notifications are enabled and disabled by configuration. IETF SNMP notifications normally define specific controls to enable and disable notifications. Events can be reported to local log as a syslog message and/or SNMP notification as defined in docsDevEvControlTable [RFC 4639] and Annex A. An EQAM supports event notification functions, including local event logging, syslog (limiting/throttling), and SNMP notification (limiting/throttling), as specified in [RFC 4639] and this specification. An EQAM is required to support SNMP trap control, as specified in [RFC 4639] and this specification.

9.2.1 Format of Events

The following sections explain in detail how to report these events by any of the three mechanisms (local event logging, SNMP notification, and syslog).

9.2.1.1 Local Event Logging

An EQAM **MUST** maintain Local Log events, defined in Annex A, in both local-volatile storage and local non-volatile storage. An EQAM **MAY** retain them in local non-volatile storage events designated for local volatile storage.

The EQAM **MUST** implement its Local Log as a cyclic buffer with a minimum of 300 entries. The EQAM Local Log non-volatile storage events **MUST** persist across reboots. The EQAM **MUST** provide access to the Local Log events through the docsDevEventTable [RFC 4639].

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

Events are identical if their EventIds are identical. For identical events occurring consecutively, the EQAM MAY choose to store only a single event. If an EQAM stores as a single event multiple identical events that occur consecutively, the EQAM MUST reflect the most recent event in the event description.

The EventId digit is a 32-bit unsigned integer. EventIds ranging [RFC 4639] from 0 to (231 - 1) are reserved by DOCSIS. The EQAM MUST report in the docsDevEvTable [RFC 4639] the EventId as a 32-bit unsigned integer, and convert the EventId from the error codes defined in Annex A to be consistent with this number format.

9.2.1.2 SNMP Notifications

An EQAM MUST implement the generic SNMP notifications according to Annex A.

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

An EQAM MUST implement SNMP notifications defined in DOCS-IF3-MIB from Annex A.

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

9.2.1.3 Syslog message format

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

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

In which:

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

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

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

9.2.2 BIT Values for docsDevEvReporting [RFC 4639]

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

- 1: local(0)
- 2: traps(1)
- 3: syslog(2)
- 4: localVolatile(8)
- 5: stdInterface(9)

Bit-0 means non-volatile Local Log storage and bit-8 are used for volatile Local Log storage (see Section 9.2.1). Bit-1 means SNMP Notifications correspond to both SNMP Trap and SNMP Inform.

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

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

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

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

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

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

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

9.2.3 Standard events for EQAM

The EQAM MUST support either volatile local log, non-volatile local log, or both.

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

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

Table 9-1 - Default event reporting mechanism versus priority (non-volatile Local Log support only)

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

Table 9-2 - Default event reporting mechanism versus priority (volatile Local Log support only)

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

Table 9–3 - Default event reporting mechanism versus priority

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

9.2.4 Event Priorities and Vendor-Specific Events

This specification defines events to most of the Event priority Levels defined by this specification. Vendor-specific events can be defined for any Event Priority Level. Table 9–4 summarizes those considerations.

Table 9–4 - Event Priorities Assignment

Event Priority	EQAM Event Assignment
Emergency	Vendor Specific
Alert	PMI and Vendor Specific (optional*)
Critical	PMI and Vendor Specific (optional*)
Error	PMI and Vendor Specific (optional*)
Warning	PMI and Vendor Specific (optional*)
Notice	PMI and Vendor Specific (optional*)
Information	PMI and Vendor Specific (optional*)
Debug	Vendor Specific

*Note: Vendor-specific optional event definitions are recommended only where the CM/CMTS allows for sufficient storage of such events.

9.3 List of events

The following list of Events are indications of error conditions at the EQAM.

Process	Sub-Process	CMTS & EQAM Priority	Event Message	Message Notes and Details	Error Code Set	Event ID	Trap Name
EQAM DHCP Provisioning							
Eqam	EQAM-PMI	Critical	CDN; Result Code = 0;	for syslog & local-log Mandatory Add: ; Error Code = 0; Optional Add: <EC Text> Should Add in <EC Text>: "; PWType = <value>"	Q01.1	81000101	

Process	Sub-Process	CMTS & EQAM Priority	Event Message	Message Notes and Details	Error Code Set	Event ID	Trap Name
Eqam	EQAM-PMI	Critical	CDN; Result Code = 0;	for syslog & local-log Mandatory Add: ; Error Code = 0; Optional Add: <EC Text> Should Add in <EC Text>: "; PWType = <value>"	Q01.2	81000102	
EQAM Configuration File Download							
Eqam	EQAM-PMI	Critical	CDN; Result Code = 0;	for syslog & local-log Mandatory Add: ; Error Code = 0; Optional Add: <EC Text> Should Add in <EC Text>: "; PWType = <value>"	Q02.1	81000201	
Eqam	EQAM-PMI	Critical	CDN; Result Code = 0;	for syslog & local-log Mandatory Add: ; Error Code = 0; Optional Add: <EC Text> Should Add in <EC Text>: "; PWType = <value>"	Q02.2	81000202	
EQAM Configuration File Processing							
Eqam	EQAM-PMI	Critical	CDN; Result Code = 0;	for syslog & local-log Mandatory Add: ; Error Code = 0; Optional Add: <EC Text> Should Add in <EC Text>: "; PWType = <value>"	E03.1	81000301	
Eqam	EQAM-PMI	Critical	CDN; Result Code = 0;	for syslog & local-log Mandatory Add: ; Error Code = 0; Optional Add: <EC Text> Should Add in <EC Text>: "; PWType = <value>"	Q03.2	81000302	

Process	Sub-Process	CMTS & EQAM Priority	Event Message	Message Notes and Details	Error Code Set	Event ID	Trap Name
EQAM ERM							
Eqam	EQAM-PMI	Critical	CDN; Result Code = 0;	for syslog & local-log Mandatory Add: ; Error Code = 0; Optional Add: <EC Text> Should Add in <EC Text>: "; PWType = <value>"	Q04.1	81000401	Session Loss (Unicast or Multicast)
Eqam	EQAM-PMI	Critical	CDN; Result Code = 0;	for syslog & local-log Mandatory Add: ; Error Code = 0; Optional Add: <EC Text> Should Add in <EC Text>: "; PWType = <value>"	Q03.2	81000302	Link Down Loss of Service
Eqam	EQAM-PMI	Critical	CDN; Result Code = 0;	for syslog & local-log Mandatory Add: ; Error Code = 0; Optional Add: <EC Text> Should Add in <EC Text>: "; PWType = <value>"	Q03.3	81000303	# of Sessions Lost & # of Sessions fail-over
Eqam	EQAM-PMI	Critical	CDN; Result Code = 0;	for syslog & local-log Mandatory Add: ; Error Code = 0; Optional Add: <EC Text> Should Add in <EC Text>: "; PWType = <value>"	Q03.4	81000304	Excessive network jitter in session, jitter buffer overflow

Annex A Detailed MIB Requirements (normative)

The abbreviations and rules in Table A–1 apply in this Annex.

Table A–1 - MIB Implementation Support

Requirement Type	Table Notation	Description
Deprecated	D	Deprecated objects are optional. If a vendor chooses to implement the object, the object must be implemented correctly according to the MIB definition. If a vendor chooses not to implement the object, an agent must not instantiate such object and must respond with the appropriate error/exception condition (e.g., 'noSuchObject' for SNMPv2c).
Mandatory	M	The object must be implemented correctly according to the MIB definition.
Not Applicable	NA	Not applicable to the device.
Not Supported	N-Sup	An agent must instantiate such object and must respond with the appropriate error/exception condition (e.g., 'noSuchObject' for SNMPv2c).
Optional	O	A vendor can choose to implement or not implement the object. If a vendor chooses to implement the object, the object must be implemented correctly according to the MIB definition. If a vendor chooses not to implement the object, an agent must not instantiate such object and must respond with the appropriate error/exception condition (e.g., 'noSuchObject' for SNMPv2c).
Obsolete	Ob	In SNMP convention, obsolete objects should not be implemented. This specification allows vendors to implement or not implement obsolete objects. If a vendor chooses to implement an obsoleted object, the object must be implemented correctly according to the MIB definition. If a vendor chooses not to implement the obsoleted object, the SNMP agent must not instantiate such object and must respond with the appropriate error/exception condition (e.g., 'noSuchObject' for SNMPv2c).
Should	S	Should. A vendor SHOULD implement the object. If a vendor implements the object, the object MUST be implemented correctly according to the MIB definition. If a vendor does not implement the object, an agent MUST NOT instantiate such object and MUST respond with the appropriate error/exception condition (e.g., no such object for SNMPv2c).

Table A–2 - SNMP Access Requirements

SNMP Access Type	Table Notation	Description
N-Acc	Not Accessible	The object is not accessible and is usually an index in a table
Read Create	RC	The access of the object MUST be implemented as Read-Create
Read Write	RW	The access of the object MUST be implemented as Read-Only
Read Only	RO	The access of the object MUST be implemented as Read-Write
Read Create or Read Only	RC/RO	The access of the object MUST be implemented as either Read-Create or Read-Only as described in the MIB definition

SNMP Access Type	Table Notation	Description
Read Write / Read Only	RW/RO	The access of the object MUST be implemented as either Read-Write or Read-Only as described in the MIB definition
Accessible for SNMP Notifications	Acc-FN	These objects are used for SNMP Notifications by the CMTS and CM SNMP Agents
Accessible through SNMP Trap	ATRAP	These objects are accessible through SNMP traps.

A.1 Object Model schema Definition

The table below lists the EQAM device compliance requirements summary.

Table A-3 - Requirements

DOCS-IF-MIB [RFC 4546]				
docsIfDownstreamChannelTable				
Objects	M-CMTS EQAM	Access	VIDEO EQAM	Access
docsIfDownChannelId	NA		NA	
docsIfDownChannelFrequency	M	RO	M	RO
docsIfDownChannelWidth	M	RO	M	RO
docsIfDownChannelModulation	M	RO	M	RO
docsIfDownChannelInterleave	M	RO	M	RO
docsIfDownChannelPower	M	RO	M	RO
docsIfDownChannelAnnex	O	RO	O	RO
SCTE-HMS-QAM-MIB [SCTE 154-2]				
qamChannelTable				
Objects	M-CMTS EQAM	Access	VIDEO EQAM	Access
qamChannelModulationFormat	O	RO	M	RO
qamChannelInterleaverLevel	O	RO	M	RO
qamChannelInterleaverMode	O	RO	M	RO
qamChannelPower	O	RO	M	RO
qamChannelSquelch	O	RO	M	RO
qamChannelContWaveMode	O	RO	M	RO
qamChannelAnnexMode	O	RO	M	RO

qamChannelCommonTable				
Objects	M-CMTS EQAM	Access	VIDEO EQAM	Access
qamChannelCommonOutputBw	O	RO	M	RO
qamChannelCommonUtilization	O	RO	M	RO
qamConfigTable				
Objects	M-CMTS EQAM	Access	VIDEO EQAM	Access
qamConfigIndex	O	N-Acc	M	N-Acc
qamConfigQamChannelIdMin	O	RO	M	RO
qamConfigQamChannelIdMax	O	RO	M	RO
qamConfigIPAddrType	O	RO	M	RO
qamConfigIPAddr	O	RO	M	RO
qamConfigUdpPortRangeMin	O	RO	M	RO
qamConfigUdpPortRangeMax	O	RO	M	RO
qamConfigOutputProgNoMin	O	RO	M	RO
qamConfigOutputProgNoMax	O	RO	M	RO
SCTE-HMS-MPEG-MIB [SCTE 154-4]				
mpegDigitalInputs				
Object	M-CMTS EQAM	Access	VIDEO EQAM	Access
mpegLossOfSignalTimeout	O	RO	M	RO
mpegInputTSTable				
Objects	M-CMTS EQAM	Access	VIDEO EQAM	Access
mpegInputTSIndex	O	N-Acc	M	N-Acc
mpegInputTSType	O	RO	M	RO
mpegInputTSConnectionType	O	RO	M	RO
mpegInputTSConnection	O	RO	M	RO
mpegInputTSPsiDetected	O	RO	M	RO
mpegInputTSStartTime	O	RO	M	RO
mpegInputTSResourceAllocated	O	RO	M	RO
mpegInputTSNumPrograms	O	RO	M	RO
mpegInputTSRate	O	RO	M	RO
mpegInputTSMAXRate	O	RO	M	RO
mpegInputTSPatVersion	O	RO	M	RO
mpegInputTSCatVersion	O	RO	M	RO
mpegInputTSNitPid	O	RO	M	RO
mpegInputTSNumEmms	O	RO	M	RO

mpegInputTSTSID	O	RO	M	RO
mpegInputTSLock	O	RO	O	RO
mpegInputProgTable				
Objects	M-CMTS EQAM	Access	VIDEO EQAM	Access
mpegInputProgIndex	O	N-Acc	M	N-Acc
mpegInputProgNo	O	RO	M	RO
mpegInputProgPmtVersion	O	RO	M	RO
mpegInputProgPmtPid	O	RO	M	RO
mpegInputProgPcrPid	O	RO	M	RO
mpegInputProgEcmPid	O	RO	M	RO
mpegInputProgNumElems	O	RO	M	RO
mpegInputProgNumEcms	O	RO	M	RO
mpegInputProgCaDescr	O	RO	M	RO
mpegInputProgScte35Descr	O	RO	O	RO
mpegInputProgScte18Descr	O	RO	O	RO
mpegProgESTable				
Objects	M-CMTS EQAM	Access	VIDEO EQAM	Access
mpegProgESIndex	O	N-Acc	M	N-Acc
mpegProgESPID	O	RO	M	RO
mpegProgESType	O	RO	M	RO
mpegProgESCaDescr	O	RO	M	RO
mpegProgESScte35Descr	O	RO	O	RO
mpegProgESScte18Descr	O	RO	O	RO
mpegInputStatsTable				
Objects	M-CMTS EQAM	Access	VIDEO EQAM	Access
mpegInputStatsPcrJitter	O	RO	M	RO
mpegInputStatsMaxPacketJitter	O	RO	M	RO
mpegInputStatsPcrPackets	O	RO	M	RO
mpegInputStatsNonPcrPackets	O	RO	M	RO
mpegInputStatsUnexpectedPackets	O	RO	M	RO
mpegInputStatsContinuityErrors	O	RO	M	RO
mpegInputStatsSyncLossPackets	O	RO	M	RO

mpegInputUdpOriginationTable				
Objects	M-CMTS EQAM	Access	VIDEO EQAM	Access
mpegInputUdpOriginationIndex	O	N-Acc	M	N-Acc
mpegInputUdpOriginationId	O	N-Acc	M	N-Acc
mpegInputUdpOriginationInetAddrType	O	RO	M	RO
mpegInputUdpOriginationSrcInetAddr	O	RO	M	RO
mpegInputUdpOriginationDestInetAddr	O	RO	M	RO
mpegInputUdpOriginationDestPort	O	RO	M	RO
mpegInputUdpOriginationActive	O	RO	M	RO
mpegInputUdpOriginationIfIndex	O	RO	M	RO
mpegInputUdpOriginationPacketsDetected	O	RO	M	RO
mpegInputUdpOriginationRank	O	RO	M	RO
mpegOutputs	O	RO	M	RO
mpegInsertPacketTable				
Objects	M-CMTS EQAM	Access	VIDEO EQAM	Access
mpegInsertPacketIndex	O	N-Acc	M	N-Acc
mpegInsertPacketListId	O	RO	M	RO
mpegInsertPacketImmediateExecution	O	RO	M	RO
mpegInsertPacketStartTime	O	RO	M	RO
mpegInsertPacketRepeat	O	RO	M	RO
mpegInsertPacketContinuousFlag	O	RO	M	RO
mpegInsertPacketRate	O	RO	M	RO
mpegInsertPacketDeviceIndex	O	RO	M	RO
mpegOutputStatsTable				
Objects	M-CMTS EQAM	Access	VIDEO EQAM	Access
mpegOutputStatsDroppedPackets	O	RO	M	RO
mpegOutputStatsFifoOverflow	O	RO	M	RO
mpegOutputStatsFifoUnderflow	O	RO	M	RO
mpegOutputStatsDataRate	O	RO	M	RO
mpegOutputStatsAvailableBandwidth	O	RO	M	RO
mpegOutputStatsChannelUtilization	O	RO	M	RO
mpegOutputStatsTotalPackets	O	RO	M	RO

mpegOutputTSTable				
Objects	M-CMTS EQAM	Access	VIDEO EQAM	Access
mpegOutputTSIndex	O	N-Acc	M	N-Acc
mpegOutputTSType	O	RO	M	RO
mpegOutputTSConnectionType	O	RO	M	RO
mpegOutputTSConnection	O	RO	M	RO
mpegOutputTSNumPrograms	O	RO	M	RO
mpegOutputTSTSID	O	RO	M	RO
mpegOutputTSNitPid	O	RO	M	RO
mpegOutputTSCaPid	O	RO	M	RO
mpegOutputTSCatInsertRate	O	RO	M	RO
mpegOutputTSPatInsertRate	O	RO	M	RO
mpegOutputTSPmtInsertRate	O	RO	M	RO
mpegOutputTSStartTime	O	RO	M	RO
mpegOutputProgTable				
Objects	M-CMTS EQAM	Access	VIDEO EQAM	Access
mpegOutputProgIndex	O	N-Acc	M	N-Acc
mpegOutputProgNo	O	RO	M	RO
mpegOutputProgPmtVersion	O	RO	M	RO
mpegOutputProgPmtPid	O	RO	M	RO
mpegOutputProgPcrPid	O	RO	M	RO
mpegOutputProgEcmPid	O	RO	M	RO
mpegOutputProgNumElems	O	RO	M	RO
mpegOutputProgNumEcms	O	RO	M	RO
mpegOutputProgCaDescr	O	RO	M	RO
mpegOutputProgScte35Descr	O	RO	O	RO
mpegOutputProgScte18Descr	O	RO	O	RO
mpegOutputProgElemStatsTable				
Objects	M-CMTS EQAM	Access	VIDEO EQAM	Access
mpegOutputProgElemStatsIndex	O	N-Acc	M	N-Acc
mpegOutputProgElemStatsPid	O	RO	M	RO
mpegOutputProgElemStatsElemType	O	RO	M	RO
mpegOutputProgElemStatsDataRate	O	RO	O	RO

mpegOutputUdpDestinationTable				
Objects	M-CMTS EQAM	Access	VIDEO EQAM	Access
mpegOutputUdpDestinationIndex	NA		NA	
mpegOutputUdpDestinationId	NA		NA	
mpegOutputUdpDestinationInetAddrType	NA		NA	
mpegOutputUdpDestinationSrcInetAddr	NA		NA	
mpegOutputUdpDestinationDestInetAddr	NA		NA	
mpegOutputUdpDestinationDestPort	NA		NA	
mpegOutputUdpDestinationIfIndex	NA		NA	
mpegProgramMappingTable				
Objects	M-CMTS EQAM	Access	VIDEO EQAM	Access
mpegProgramMappingIndex	O	N-Acc	M	N-Acc
mpegProgramMappingOutputProgIndex	O	RO	M	RO
mpegProgramMappingOutputTSIndex	O	RO	M	RO
mpegProgramMappingInputProgIndex	O	RO	M	RO
mpegProgramMappingInputTSIndex	O	RO	M	RO
mpegVideoSessionTable				
Objects	M-CMTS EQAM	Access	VIDEO EQAM	Access
mpegVideoSessionIndex	O	N-Acc	M	N-Acc
mpegVideoSessionPhyMappingIndex	O	RO	M	RO
mpegVideoSessionPIDRemap	O	RO	M	RO
mpegVideoSessionMode	O	RO	M	RO
mpegVideoSessionState	O	RO	M	RO
mpegVideoSessionProvMethod	O	RO	M	RO
mpegVideoSessionEncryptionType	O	RO	M	RO
mpegVideoSessionEncryptionInfo	O	RO	M	RO
mpegVideoSessionBitRate	O	RO	M	RO
mpegVideoSessionID	O	RO	M	RO
mpegVideoSessionSelectedInput	O	RO	M	RO
mpegVideoSessionSelectedOutput	O	RO	M	RO

mpegVideoSessionPtrTable				
Objects	M-CMTS EQAM	Access	VIDEO EQAM	Access
mpegVideoSessionPtrInputProgIndex	O	N-Acc	M	N-Acc
mpegVideoSessionPtrInputTSIndex	O	RO	M	RO
mpegVideoSessionPtrInputTSConnType	O	RO	M	RO
mpegVideoSessionPtrInputTSConnection	O	RO	M	RO
mpegVideoSessionPtrOutputProgIndex	O	RO	M	RO
mpegVideoSessionPtrOutputTSIndex	O	RO	M	RO
mpegVideoSessionPtrOutputTSConnType	O	RO	M	RO
mpegVideoSessionPtrOutputTSConnection	O	RO	M	RO
mpegVideoSessionPtrStatus	O	RO	M	RO
DOCS-DRF-MIB [SCTE 137-3]				
docsDrfDownstreamTable				
Object	M-CMTS EQAM	Access	VIDEO EQAM	Access
docsDrfDownstreamPhyDependencies	NA		NA	
docsDrfDownstreamCapabTable				
Objects	M-CMTS EQAM	Access	VIDEO EQAM	Access
docsIfMCmtsEqamDownstreamCapabFrequency	M	RO	O	RO
docsIfMCmtsEqamDownstreamCapabBandwidth	M	RO	O	RO
docsIfMCmtsEqamDownstreamCapabPower	M	RO	O	RO
docsIfMCmtsEqamDownstreamCapabModulation	M	RO	O	RO
docsIfMCmtsEqamDownstreamCapabInterleaver	M	RO	O	RO
docsIfMCmtsEqamDownstreamCapabJ83Annex	M	RO	O	RO
docsIfMCmtsEqamDownstreamCapabConcurrentServices	M	RO	O	RO
docsIfMCmtsEqamDownstreamCapabServicesTransport	M	RO	O	RO
docsIfMCmtsEqamDownstreamCapabMuting	M	RO	O	RO
docsDrfGroupDependencyTable				
Objects	M-CMTS EQAM	Access	VIDEO EQAM	Access
docsDrfGroupDependencyPhyParam	M	N-Acc	O	N-Acc
docsDrfGroupDependencyPhysicalIndex	M	N-Acc	O	N-Acc
docsDrfGroupDependencyGroupID	M	RO	O	RO
docsDrfGroupDependencyType	M	RO	O	RO