

# Superseded

## **Data-Over-Cable Service Interface Specifications**

### **Operations Support System Interface Specification**

**SP-OSSlv1.1-I02-000714**

**INTERIM  
SPECIFICATION**

#### **Notice**

This document is a cooperative effort undertaken at the direction of Cable Television Laboratories, Inc. for the benefit of the cable industry in general. Neither CableLabs nor any member company is responsible for any liability of any nature whatsoever resulting from or arising out of use or reliance upon this specification by any party. This document is furnished on an "AS IS" basis and neither CableLabs nor its members provides any representation or warranty, express or implied, regarding its accuracy, completeness, or fitness for a particular purpose. Distribution of this document is restricted pursuant to the terms of separate access agreements negotiated with each of the parties to whom this document has been furnished.

© Copyright 1999, 2000 Cable Television Laboratories, Inc.  
All rights reserved.

## Document Status Sheet

Document Control Number: SP-OSSiv1.1-I02-000714

Revision History: I01 – First Interim Release, April 7, 2000  
I012– Incorporated ECN OSS-N-00039, OSS-N-00040, OSS-N-00041, OSS-N-00054, July 14, 2000

Date: July 14, 2000

**Status Code:**      ~~Work in Process~~      ~~Draft~~      Interim      ~~Released~~

**Distribution Restrictions:**      ~~CableLabs® & Members Only~~      ~~CableLabs®, Members, and Vendors Only~~      Public

## Key to Document Status Codes

Work in Process	An incomplete document, designed to guide discussion and generate feedback, that may include several alternative requirements for consideration.
Draft	A document in specification format considered largely complete, but lacking review by Members and vendors. Drafts are susceptible to substantial change during the review process.
Interim	A document which has undergone rigorous review by Members and vendors, suitable for use by vendors to design in conformance with, and suitable for field testing.
Released	A stable document, reviewed, tested and validated, suitable to enable cross-vendor interoperability.

CableLabs® is a registered trademark of Cable Television Laboratories, Inc.

## Table of Contents

TABLE OF CONTENTS .....	3
<b>1 SCOPE AND PURPOSE.....</b>	<b>4</b>
1.1 SCOPE.....	4
1.2 REQUIREMENTS .....	4
<b>2 SNMP PROTOCOL .....</b>	<b>5</b>
2.1 SNMP MODE FOR DOCSIS 1.1 COMPLIANT CMTS .....	5
2.1.1 Key Change Mechanism .....	6
2.2 SNMP MODE FOR DOCSIS 1.1 COMPLIANT CMS .....	6
2.2.1 SNMPv3 Initialization and Key changes .....	7
2.2.2 SNMPv3 Initialization.....	7
2.2.3 DH Key Changes .....	10
2.2.4 VACM Profile .....	10
<b>3 MANAGEMENT INFORMATION BASES (MIBS) .....</b>	<b>13</b>
3.1 IPCDN DRAFTS AND OTHERS .....	13
3.2 IETF RFCs .....	14
3.3 MANAGED OBJECTS REQUIREMENTS.....	14
3.3.1 CMTS MIB requirements.....	14
3.3.2 Requirements for RFC-2669.....	14
3.3.3 Requirements for RFC-2670.....	14
3.3.4 Requirements for RFC-2233.....	15
3.3.5 Interface MIB and Trap Enable.....	17
3.3.6 Requirements for RFC-2665.....	17
3.3.7 Requirements for RFC-1493.....	17
3.3.8 Requirements for RFC-2011.....	17
3.3.9 Requirements for RFC-2013.....	18
3.3.10 Requirements for RFC-1907.....	18
3.3.11 Requirements for “draft-ietf-ipcdn-qos-mib-02.txt” .....	18
3.3.12 Requirements for “draft-ietf-ipcdn-igmp-mib-01.txt” .....	18
3.3.13 Requirements for “draft-ietf-idmr-igmp-mib-13.txt” .....	18
3.3.14 Requirements for “draft-ietf-ipcdn-bpiplus-mib-02.txt” BPI+ MIB.....	18
3.3.15 Requirements for “draft-ietf-xxxx-xxxx-xxxx-00.txt” USB MIB.....	18
3.3.16 Requirements for “draft-ietf-ipcdn-subscriber-mib-01.txt” Subscriber Management MIB.....	18
3.3.17 Requirements for RFC-2786 Diffie-Helman USM Key.....	19
3.4 CM CONFIGURATION FILES, TLV-11 AND MIB OIDS/VALUES.....	19
3.4.1 CM configuration file TLV-11 element translation (to SNMP PDU) .....	19

3.4.2	Ignore CM configuration TLV-11 elements which are not supported by CM.....	20
3.4.3	CM state after CM configuration file processing success.....	20
3.4.4	CM state after CM configuration file processing failure.....	20
<b>4</b>	<b>OSSI FOR RADIO FREQUENCY INTERFACE.....</b>	<b>21</b>
4.1	SUBSCRIBER ACCOUNT MANAGEMENT INTERFACE SPECIFICATION.....	21
4.2	CONFIGURATION MANAGEMENT.....	21
4.2.1	Version Control.....	21
4.2.2	System Initialization and Configuration.....	22
4.2.3	Secure Software Upgrades.....	22
4.3	PROTOCOL FILTERS.....	26
4.3.1	LLC Filter.....	26
4.3.2	Special Filter.....	27
4.3.3	IP Spoofing Filter.....	27
4.3.4	SNMP Access Filter.....	28
4.3.5	IP Filter.....	29
4.4	FAULT MANAGEMENT.....	29
4.4.1	SNMP Usage.....	29
4.4.2	Event Notification.....	30
4.4.3	Trap and Syslog Throttling, Trap and Syslog Limiting.....	35
4.4.4	Non-SNMP Fault Management Protocols.....	35
4.5	PERFORMANCE MANAGEMENT.....	35
4.5.1	Additional MIB Implementation Requirements.....	36
4.6	COEXISTENCE.....	37
	Coexistence and MIBs.....	37
4.6.2	Coexistence and SNMP.....	39
<b>5</b>	<b>OSS FOR BPI+ .....</b>	<b>40</b>
<b>6</b>	<b>OSSI FOR CMCI.....</b>	<b>41</b>
6.1	SNMP ACCESS VIA CMCI.....	41
6.2	CONSOLE ACCESS.....	41
6.3	CM DIAGNOSTIC CAPABILITIES.....	42
6.4	PROTOCOL FILTERING.....	42
6.5	MANAGEMENT INFORMATION BASE (MIB) REQUIREMENTS.....	42
	<b>APPENDIX A. DETAILED MIB REQUIREMENTS .....</b>	<b>43</b>
	<b>APPENDIX B. BUSINESS PROCESS SCENARIOS FOR SUBSCRIBER ACCOUNT MANAGEMENT ...</b>	<b>74</b>
B.1.	THE OLD SERVICE MODEL -- “ONE CLASS ONLY” & “BEST EFFORT” SERVICE .....	74
B.2.	THE OLD BILLING MODEL -- “FLAT RATE” ACCESS .....	74
B.3.	A SUCCESSFUL NEW BUSINESS PARADIGM.....	74
B.3.1	Integrating “Front End” Processes Seamlessly with “Back Office” Functions.....	75
B.3.2	Designing Class of Services.....	75
B.3.3	Usage-Based Billing.....	76
B.3.4	Designing Usage-Based Billing Models.....	76
	<b>APPENDIX C. PROPOSE ACCOUNT MANAGEMENT MIB .....</b>	<b>78</b>
	<b>APPENDIX D. SNMPV2C INFORM REQUEST DEFINITION FOR SUBSCRIBER ACCOUNT</b>	
	<b>MANAGEMENT (SAM).....</b>	<b>79</b>

<b>APPENDIX E. SUMMARY OF THE CM AUTHENTICATION AND THE CODE FILE AUTHENTICATION.....</b>	<b>80</b>
E.1    AUTHENTICATION OF THE DOCSIS 1.1 COMPLIANT CM .....	80
E.1.1. <i>Responsibility of the DOCSIS Root CA.....</i>	<i>81</i>
E.1.2 <i>Responsibility of the CM manufacturers.....</i>	<i>81</i>
E.1.3 <i>Responsibility of the operators .....</i>	<i>81</i>
E.2    AUTHENTICATION OF THE CODE FILE FOR THE DOCSIS 1.1 COMPLIANT CM .....	82
E.2.1 <i>Responsibility of the DOCSIS Root CA.....</i>	<i>83</i>
E.2.2 <i>Responsibility of the CM manufacturer .....</i>	<i>83</i>
E.2.3 <i>Responsibility of CableLabs .....</i>	<i>83</i>
E.2.4 <i>Responsibility of the operators .....</i>	<i>83</i>
<b>APPENDIX F. EVENTS FOR NOTIFICATION.....</b>	<b>85</b>
<b>APPENDIX G. TRAP DEFINITIONS FOR CABLE DEVICE.....</b>	<b>94</b>
<b>APPENDIX H. REFERENCES .....</b>	<b>95</b>
<b>APPENDIX I. ACKNOWLEDGEMENTS .....</b>	<b>99</b>

# 1 Scope and Purpose

## 1.1 Scope

This Document specifies the Network Management requirements for supporting DOCSIS 1.1 environment. More specifically, the specification details SNMPv3 protocol and its compatibility with MP V. The RFCs and Management Information Base (MIB) requirements are detailed as reference for testing, event notification, etc. The network management principals such as access, configuration, fault and performance management are incorporated in this specification for better understanding of managing a high-speed cable modem environment.

## 1.2 Requirements

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

“MUST”	This word or the adjective “REQUIRED” means that the item is an absolute requirement for this specification.
“MUST NOT”	This phrase means that the item is an absolute prohibition of this specification.
“SHOULD”	This word or the adjective “RECOMMENDED” 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 weighted 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 weighted before implementing any behavior described with this label.
“MAY”	This word or the adjective “OPTIONAL” means that this item is truly optional. One vendor may choose to include the item because a particular marketplace required it or because it enhances the product, for example; another vendor may omit the same item.

## 2 SNMP Protocol

The SNMPV3 protocol has been selected as the communication protocol for management of data-over-cable Services and MUST be implemented. Although SNMPv3 offers advantages, many management systems may not be capable of supporting SNMPV3 agents. Thus, support of SNMPv1 and SNMPv2c is also required and MUST be implemented.

The following IETF SNMP related RFCs MUST be implemented:

RFC-2570	Introduction to Version 3 of the internet-standard Network Management
RFC-2571	An Architecture for Describing SNMP Management Frameworks
RFC-2572	Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)
RFC-2573	SNMP Applications
RFC-2574	User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)
RFC-2575	View-based Access Control Model (VACM) for the simple Network Management Protocol (SNMP)
RFC-1905	Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)
RFC-1906	Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)
RFC-1907	Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)
RFC-1901	Introduction to Community-based SNMPv2
RFC-1157	A Simple Network Management Protocol

For support of SMIPv2 the following IETF SNMP related RFC's MUST be implemented:

RFC-2578	Structure of Management Information Version 2 (SMIPv2)
RFC-2579	Textual Conventions for SMIPv2
RFC-2580	Conformance Statements for SMIPv2
RFC-2786	Diffie-Helman USM Key

### 2.1 SNMP Mode for DOCSIS 1.1 compliant CMTS

DOCSIS 1.1 compliant CMTS must support both SNMPv1/SNMPv2c and SNMPv3.

DOCSIS 1.1 compliant CMTS MUST operate in one of the two modes:

- V1/V2c mode
- V3 mode

The CMTS is in V1/V2c mode unless it is configured for SNMPv3.

It is up to the vendor to provide a mechanism to put DOCSIS 1.1 compliant CMTS into SNMPv1/SNMPv2c or SNMPv3 operational state.

DOCSIS 1.1 compliant CMTS MAY support kick-start mechanism as specified in the RFC-2786.

DOCSIS 1.1 compliant CMTS MUST NOT implement SNMP coexistence as described by [RFC-2576]

When CMTS is in V1/V2 mode, it MUST:

- allow RO/RC/RW operation to V1 packets
- allow RO/RC/RW operation to V2c packets
- not allow RO/RC/RW operation to V3 packets

When CMTS is in V3 mode, it:

- MUST allow RO/RC/RW operation to V3 packets
- MUST NOT allow RW/RC/RO operation to V2c packets.
- MUST NOT allow RW/RC/RO operation to V1 packets.

Note:

RO = read only

RC = read-create

RW = read-write

### **2.1.1 Key Change Mechanism**

DOCSIS 1.1 compliant CMTS SHOULD use the key-change mechanism specified in the RFC-2786. CMTS MUST always support the key-change mechanism described in the RFC-2574 to comply with industry wide SNMP V3 standard.

## **2.2 SNMP Mode for DOCSIS 1.1 compliant CMs**

DOCSIS 1.1 compliant CMs (in 1.1 and 1.0 mode) must support both SNMPv1/SNMPv2c and SNMPv3.

DOCSIS 1.1 compliant CM MUST operate in one of the two modes:

- V1/V2c mode
- V3 mode

The modem is in V1/V2c mode unless it sees appropriate snmpv3-kickstart TLVs in the configuration file as specified in RFC-2786 and [MCNS5]

DOCSIS 1.1 compliant CM MUST NOT implement SNMP coexistence as described by [RFC-2576]

When CM is in V1/V2 mode, it MUST:

- allow RO/RC/RW operation to V1 packets
- allow RO/RC/RW operation to V2c packets
- not allow RO/RC/RW operation to V3 packets



When CM is in V3 mode, it:

- MUST allow RO/RC/RW operation to V3 packets
- MUST NOT allow RW/RC/RO operation to V2c packets.
- MUST NOT allow RW/RC/RO operation to V1 packets.

Note:

RO = read only

RC = read-create

RW = read-write

## 2.2.1 SNMPv3 Initialization and Key changes

DOCSIS 1.1 compliant CM MUST support the “SNMPv3 Initialization” and “DH Key Changes” requirements specified in the following sections.

### 2.2.2 SNMPv3 Initialization

1. For each of up to 5 different security names, the Manager generates a pair of numbers:

- a. Manager generates a random number  $R_m$
- b. Manager uses DH equation to translate  $R_m$  to a public number  $z$

$z = g^{R_m} \text{ MOD } p$  where  $g$  is the from the set of Diffie-Hellman parameters,  $p$  is the prime from those parameters

2. CM configuration file is created to include (security name, public number) pair. CM MUST support a minimum of 5 pairs.

**For example:**

**TLV type 34.1 (SnmpV3 Kickstart Security Name) = docsisManager**

**TLV type 34.2 (SnmpV3 Kickstart Public Number) =  $z$**

CM MUST support VACM entries defined in section 2.3 “VACM Profile”. Each of these entries will only be active if the corresponding security name is specified in the CM configuration file.

During the CM boot up process, the above values will be populated in the `usmDhKickstartTable`.

At this point:

`usmDhKickstartMgrpublic.1 = “z”` (octet string)

`usmDhKickstartSecurityName.1 = “docsisManager”`

When `usmDhKickstartMgrpublic.n` is set with a valid value during the registration, a corresponding row is created in the `usmUserTable` with the following values:

usmUserEngineID	localEngineID
usmUserName	usmDHKickstartSecurityName.n value
usmuserSecurityName	usmDHKickstartSecurityName.n value
usmUserCloneForm	ZeroDotZero
usmUserAuthProtocol	usmHMACMD5AuthProtocol
usmuserAuthKeyChange	-- derived from set value
usmUserOwnAuthKeyChange	-- derived from set value
usmUserPrivProtocol	usmDESPrivProtocol
usmUserPrivKeyChange	-- derived from set value
usmUserOwnPrivKeyChange	-- derived from set value
usmUserPublic	“
usmUserStorageType	permanent
usmUserStatus	active

**Note:** For dhKickstart entries in usmUserTable, Permanent means it can be written to but not deleted and is not saved across reboots.

After the CM has registered with the CMTS.

1) CM generates a random number  $x_a$  for each row populated in the usmDHKickstartTable which has a non zero length usmDHKickstartSecurityName and usmDHKickstartMgrPublic.

2) CM uses DH equation to translate  $x_a$  to a public number  $c$  (for each row identified above)

$c = g^{x_a} \text{ MOD } p$  where  $g$  is the from the set of Diffie-Helman parameters,  $p$  is the prime from those parameters

At this point:

usmDHKickstartMyPublic.1 = “c” (octet string)

usmDHKickstartMgrPublic.1 = “z” (octet string)

usmDHKickstartSecurityName.1 = “docsisManager”

3) CM calculate shared secret  $sk$  where  $sk = z^{x_a} \text{ mod } p$

4) CM uses  $sk$  to derive the privacy key and authentication key for each row in usmDHKickstartTable and sets the values into the usmUserTable

As specified in RFC-2786, the privacy key and the authentication key for the associated username, “docsisManager” in this case, is derived from  $sk$  by applying the key derivation function PBKDF2 defined in PKCS#5v2.0.

```
privacy key <--- PBKDF2( salt = 0xd1310ba6,  
iterationCount = 500,  
keyLength = 16,  
prf = id-hmacWithSHA1)  
  
authentication key <---- PBKDF2( salt = 0x98dfb5ac,  
iterationCount = 500,  
keyLength = 16 (usmHMACMD5AuthProtocol),  
prf = id-hmacWithSHA1)
```

At this point the CM has completed its SNMPv3 initialization process and will allow appropriate access level to a valid securityname with the correct authentication key and/or privacy key.

DOCSIS 1.1 compliant CM MUST properly populate keys to appropriate tables as specified by the SNMPv3 related RFCs and RFC-2786.

In case of failure to complete SNMP V3 initialization, the modem MUST enter SNMP V1/V2 compatibility mode. The contents of the docsDevNmAccessTable MUST then control access. The usmDHKickstartTable MUST read as an empty table to indicate the modem is in SNMP V1/V2 compatibility mode

The following describes the process that the manager uses to derive CM's unique authentication key and privacy key.

5) SNMP manager accesses the contents of the usmDHKickstartTable using the security name of 'dhKickstart' with no authentication.

DOCSIS 1.1 compliant CM MUST provide preinstalled entries in the USM table and VACM tables to correctly create user 'dhKickstart' of security level noAuthnoPriv that has read only access to system group and usmDHkickstartTable.

SNMP manager gets the value of CM's usmDHKickstartMypublic number associated with the securityname that manager wants to derive authentication and privacy keys for. With the manager's knowledge of the private random number, the manager can calculate the DH shared secret. From that shared secret, the manager can derive operational authentication and confidentiality keys for the securityname that the manager is going to use to communicate with the CM.

### 2.2.3 DH Key Changes

DOCSIS 1.1 compliant CM MUST support the key-change mechanism specified in the RFC-2786.

### 2.2.4 VACM Profile

This section will address the VACM profile for DOCSIS CM when it is operating in SNMPv3 OSS environment.

In addition to RFC-2575, the following VACM entries MUST be included by default in a compliant CM:

-- The system manager, has full read/write/config access

vacmSecurityModel	3 (USM)
vacmSecurityName	'docsisManager'
vacmGroupName	'docsisManager'
vacmSecurityToGroupStorageType	permanent
vacmSecurityToGroupStatus	active

-- An operator/CSR. Has read/reset access to full modem

vacmSecurityModel	3 (USM)
vacmSecurityName	'docsisOperator'
vacmGroupName	'docsisOperator'
vacmSecurityToGroupStorageType	permanent
vacmSecurityToGroupStatus	active

-- RF Monitoring. Has read access to RF plant statistics

vacmSecurityModel	3 (USM)
vacmSecurityName	'docsisMonitor'
vacmGroupName	'docsisMonitor'
vacmSecurityToGroupStorageType	permanent
vacmSecurityToGroupStatus	active

-- User debugging. Has read access to 'useful' variables.

vacmSecurityModel	3 (USM)
vacmSecurityName	'docsisUser'
vacmGroupName	'docsisUser'
vacmSecurityToGroupStorageType	permanent
vacmSecurityToGroupStatus	active

-- Group name to view translations

vacmGroupName	'docsisManager'
vacmAccessContextPrefix	"
vacmAccessSecurityModel	3 (USM)
vacmAccessSecurityLevel	AuthPriv
vacmAccessContextMatch	exact
vacmAccessReadViewName	'docsisManagerView'
vacmAccessWriteViewName	'docsisManagerView'
vacmAccessNotifyViewName	'docsisManagerView'
vacmAccessStorageType	permanent
vacmAccessStatus	active
vacmGroupName	'docsisOperator'
vacmAccessContextPrefix	"
vacmAccessSecurityModel	3 (USM)
vacmAccessSecurityLevel	AuthPriv & AuthNoPriv
vacmAccessContextMatch	exact
vacmAccessReadViewName	'docsisManagerView'
vacmAccessWriteViewName	'docsisOperatorWriteView'
vacmAccessNotifyViewName	'docsisManagerView'
vacmAccessStorageType	permanent
vacmAccessStatus	active
vacmGroupName	'docsisMonitor'
vacmAccessContextPrefix	"
vacmAccessSecurityModel	3 (USM)
vacmAccessSecurityLevel	AuthNoPriv
vacmAccessContextMatch	exact
vacmAccessReadViewName	'docsisMonitorView'
vacmAccessWriteViewName	"
vacmAccessNotifyViewName	'docsisMonitorView'
vacmAccessStorageType	permanent
vacmAccessStatus	active
vacmGroupName	'docsisUser'
vacmAccessContextPrefix	"
vacmAccessSecurityModel	3 (USM)
vacmAccessSecurityLevel	AuthNoPriv
vacmAccessContextMatch	exact
vacmAccessReadViewName	'docsisUserView'
vacmAccessWriteViewName	"

vacmAccessNotifyViewName	"
vacmAccessStorageType	permanent
vacmAccessStatus	active

-- The views.

docsisManagerView

    subtree 1.3.6.1 (Entire mib).

docsisOperatorWriteView

    subtree 'docsDevBase'

    subtree 'docsDevSoftware'

    subtree 'docsDevEvControl'

    subtree 'docsDevEvThrottleAdminStatus'

docsisMonitorView

    subtree 1.3.6.1.2.1.1 (system)

    subtree 'docsIfBaseObjects'

    subtree 'docsIfCmObjects'

docsisUserView

    subtree 1.3.6.1.2.1.1 (system)

    subtree 'docsDevBase'

    subtree 'docsDevSwOperStatus'

    subtree 'docsDevSwCurrentVersion'

    subtree 'docsDevServerConfigFile'

    subtree 'docsDevEventTable'

    subtree 'docsDevCpeTable'

    subtree 'docsIfUpstreamChannelTable'

    subtree 'docsIfDownstreamChannelTable'

    subtree 'docsIfSignalQualityTable'

    subtree 'docsIfCmStatusTable'

DOCSIS 1.1 compliant CM MUST also support additional VACM users as they are configured via an SNMP-embedded configuration file.

### 3 Management Information Bases (MIBs)

This section defines the minimum set of managed objects required to support the management of CM and CMTS. Vendors MAY augment this MIB with objects from other standard or vendor-specific MIBs where appropriate.

DOCSIS OSSl 1.1 specification has priority over IETF MIB specification. Vendor MUST implement MIB requirements in accordance with the texts specified in OSSl 1.1 specification. Certain objects are deprecated or obsolete but may be required by the OSSl specification as mandatory and MUST be implemented.

1. Deprecated object. It is optional. That is, 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., no such object for SNMPv2c)
2. Optional object. 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., no such object for SNMPv2c)
3. Obsolete object. It is optional. 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., no such object for SNMPv2c)

Section 3.1 and 3.2 include an overview of the MIB modules required for the management of the facilities specified in SP-RFI-1.1 and BPI+ specifications.

#### 3.1 IPCDN Drafts and Others

MIB	Applicable Device(s)
IETF Proposed Standard RFC version of Qos MIB, “draft-ietf-ipcdn-qos-mib-02.txt”	CM and CMTS
IETF Proposed Standard RFC version of IGMP Proxy MIB, “draft-ietf-ipcdn-igmp-mib-01.txt”	CM and CMTS
IETF Proposed Standard RFC version of IGMP MIB, “draft-ietf-idmr-igmp-mib-13.txt”	CM and CMTS
IETF Proposed Standard RFC version of BPI+ MIB, “draft-ietf-ipcdn-bpiplus-mib-02.txt”	CM and CMTS
IETF Proposed Standard RFC version of USB MIB, “draft-ietf-xxxx-xxxx-xxxx-00.txt”	CM only
IETF Proposed Standard RFC version of BPI MIB, “draft-ietf-ipcdn-mcns-bpi-mib-01.txt”	CM and CMTS
IETF Proposed Standard RFC version of Subscriber Management MIB, “draft-ietf-ipcdn-subscriber-mib-01.txt”	CMTS only

## 3.2 IETF RFCs

MIB	Applicable Device(s)
RFC-2669: DOCSIS Cable Device MIB	CM and CMTS
RFC-2670: Radio Frequency (RF) Interface MIB	CM and CMTS
RFC-2665: Ethernet Interface MIB.	CM and CMTS
RFC-2233: The Interfaces Group MIB using SMIPv2	CM and CMTS
RFC-1493: Bridge MIB	CM and CMTS
RFC-2011: SNMPv2 Management Information Base for the Internet Protocol using SMIPv2	CM and CMTS
RFC-2013: SNMPv2 Management Information Base for the User Datagram Protocol using SMIPv2	CM and CMTS
RFC-1907: Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)	CM and CMTS
RFC-2786: Diffie-Helman USM Key	CM and CMTS

## 3.3 Managed Objects Requirements

The following sections detail any additional implementation requirements for the RFCs listed. Reference Appendix A for specific object implementation requirements.

### 3.3.1 CMTS MIB requirements

DOCSIS 1.1 compliant CMTS MUST implement Subscribe Management MIB.

### 3.3.2 Requirements for RFC-2669

RFC-2669 MUST be implemented by DOCSIS 1.1 compliant CMs. DOCSIS 1.1 compliant CMTS MUST implement mandatory required objects (as specify by Appendix A), and SHOULD implement the other non-mandatory required objects.

### 3.3.3 Requirements for RFC-2670

RFC-2670 MUST be implemented by DOCSIS 1.1 compliant CMTS and CMs.

The docsIfDownChannelPower object-type MUST be implemented in a CMTS that provides an integrated RF upconverter. If the CMTS relies on an external upconverter, then the CMTS SHOULD implement the docsIfDownChannelPower object-type. The CMTS transmit power reported in the MIB object MUST be within 2 dB of the actual transmit power in dBmV when implemented. IF transmit power management is not implemented, the MIB object will be read-only and report the value of 0 (zero).



The docsIfDownChannelPower object-type MUST be implemented in DOCSIS 1.1 conforming CM's. This object is read-only. When operated at nominal line voltage, at normal room temperature, the reported power MUST be within 3 dB of the actual received channel power. Across the input power range from -15 dBmV to +15 dBmV, for any 1 dB change in input power, the CM MUST report a power change in the same direction that is not less than 0.5 dB, and not more than 1.5 dB.

The access of docsIfDownChannelFrequency object MUST be implemented as RW if a CMTS is in control of the downstream frequency. But if a CMTS provides IF output, docsIfDownChannelFrequency MUST be implemented as read-only and return 0.

### 3.3.4 Requirements for RFC-2233

RFC-2233 MUST be implemented by DOCSIS 1.1 compliant CMTS and CMs.

The ifAdminStatus object MUST provide administrative control over both MAC interfaces and individual channel, MUST be implemented as RW.

The ifType object has been assigned the following enumerated values for each instance of a Data Over Cable Service (DOCS) interface:

CATV MAC interface:	docsCableMacLayer (127)
CATV downstream channel:	docsCableDownstream (128)
CATV upstream channel:	docsCableUpStream (129)

#### 3.3.4.1 Interface Organization and Numbering

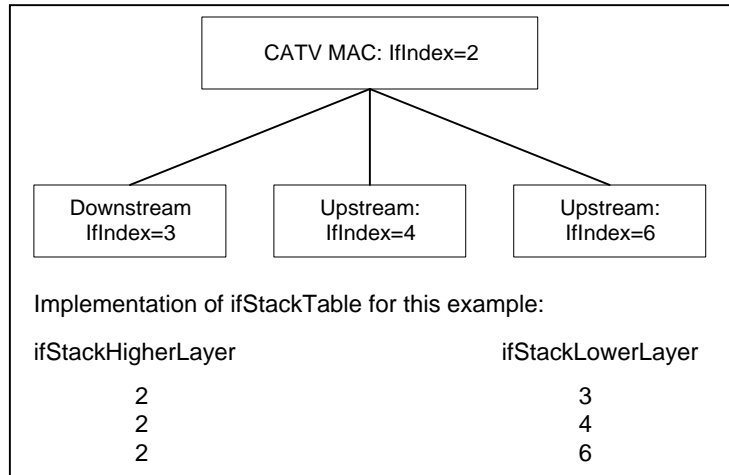
Assigned interface numbers for CATV-MAC and Ethernet (Ethernet-like interface) are used in both the NMAccessTable and IP/LLC filtering table to configure access and traffic policy at these interfaces. These configurations are generally encoded in the configuration file using TLV encoding. To avoid provisioning complexity the interface-numbering scheme MUST comply with the following requirements:

An instance of IfEntry MUST exist for each CATV-MAC interface, downstream channel, upstream channel, and LAN interface.

If a MAC interface consists of more than one upstream and downstream channel, then a separate instance of ifEntry MUST also exist for each channel.

The ifStack group ([RFC-2233]) must be implemented to identify relationship among sub-interfaces. Note that the CATV-MAC interface MUST exist, even though it is broken out into sub-interfaces.

The example below illustrates a MAC interface with one downstream and two upstream channels for a CMTS.



**Figure 1: Ifindex Example for CMTS**

At the CMTS, interface number is at the discretion of the vendor, and SHOULD correspond to the physical arrangement of connections. If table entries exist separately for upstream and downstream channels, then the ifStack group ([RFC-2233]) MUST be implemented to identify the relationship among sub-interfaces. Note that the CATV MAC interface(s) MUST exist, even if further broken out into sub-interfaces.

At the CM, interface MUST be numbered as:

Interfaces	Type
1	primary CPE interface
2	CATV-MAC
3	RF-down
4	RF-Up
4+n	Other interfaces

If the CM has only one CPE interface (in most cases it is true), then such interface is the primary CPE interface. If CM has more than one CPE interface, then the vendor MUST define which of (n) CPE interfaces is the primary CPE interface. Regardless, the Primary CPE interface MUST be Interface number 1.

The secondary CPE, and other interfaces, will start at 5.

DOCSIS CM may have multiple interfaces. If filter(s) are applied to IfIndex 1, then the filter(s) are also applied to each CPE interface; however, filters are never used to limit traffic between CM CPE interfaces.

### 3.3.5 Interface MIB and Trap Enable

Interface MIB and Trap Enable specified in RFC-2233 MUST be implemented by DOCSIS 1.1 compliant CMTS and CMs.

If a multi-layer interface model is present in the device, each sub-layer for which there is an entry in the ifTable can generate linkUp/Down traps. Since interface state changes would tend to propagate through the interface stack (from top to bottom, or bottom to top), it is likely that several traps would be generated for each linkUp/Down occurrence. The CM and CMTS MUST implement the ifLinkUpDownTrapEnable object to allow managers to control trap generation, and configure only the interface sub-layers of interest.

The default setting of ifLinkUpDownTrapEnable MUST limit the number of traps generated to one, per interface, per linkUp/Down event. Interface state changes, of most interest to network managers, occur at the lowest level of an interface stack.

On CM linkUp/Down event a trap SHOULD be generated by the CM MAC interface and not by any sub-layers of the interface. Therefore, the default setting of ifLinkUpDownTrapEnable for CM MAC MUST be set to enable, and the default setting of ifLinkUpDownTrapEnable for CM RF-Up MUST be set to disable, and the default setting of ifLinkUpDownTrapEnable for CM RF-Down MUST be set to disable.

On CMTS interfaces (MAC, RF-Downstream(s), RF-Upstream(s)) the linkUp/Down event/trap SHOULD be generated by each CMTS interface. Therefore, the default setting of ifLinkUpDownTrapEnable for each CMTS interface (MAC, RF-Downstream(s), RF-Upstream(s)) MUST be set to enable.

### 3.3.6 Requirements for RFC-2665

RFC-2665 MUST be implemented by DOCSIS 1.1 compliant CMTS and CM if Ethernet or Fast Ethernet interfaces are present.

### 3.3.7 Requirements for RFC-1493

RFC-1493 MUST be implemented by DOCSIS 1.1 compliant CMTS and CMs.

In both the CM and the CMTS (if the CMTS implements transparent bridging), the Bridge MIB ([RFC-1493]) MUST be implemented to manage the bridging process.

In the CMTS that implements transparent bridging, the Bridge MIB MUST be used to represent information about the MAC Forwarder states.

### 3.3.8 Requirements for RFC-2011

RFC-2011 MUST be implemented by DOCSIS 1.1 compliant CMTS and CMs.

#### 3.3.8.1 The IP Group

The IP group MUST be implemented. It does not apply to IP packets forwarded by the device as a link-layer bridge. For the CM, it applies only to the device as an IP host. At the CMTS, it applies to the device as an IP host, and as a routers if IP routing is implemented.

### **3.3.8.2 The ICMP Group**

The ICMP group MUST be implemented.

### **3.3.9 Requirements for RFC-2013**

RFC-2013 MUST be implemented by DOCSIS 1.1 compliant CMTS and CMs.

The UDP group in the RFC-2013 MUST be implemented.

### **3.3.10 Requirements for RFC-1907**

RFC-1907 MUST be implemented by DOCSIS 1.1 compliant CMTS and CMs.

#### **3.3.10.1 The System Group**

The System Group from RFC-1907 MUST be implemented. See Section 4.2.1 for sysObjectID requirements.

#### **3.3.10.2 The SNMP Group**

The SNMP Group from RFC-1907 MUST be implemented.

### **3.3.11 Requirements for “draft-ietf-ipcdn-qos-mib-02.txt”**

“draft-ietf-ipcdn-qos-mib-02.txt” MUST be implemented by DOCSIS 1.1 compliant CMTS and CMs.

### **3.3.12 Requirements for “draft-ietf-ipcdn-igmp-mib-01.txt”**

“draft-ietf-ipcdn-igmp-mib-01.txt” MUST be implemented by DOCSIS 1.1 compliant CMTS and CMs.

### **3.3.13 Requirements for “draft-ietf-idmr-igmp-mib-13.txt”**

“draft-ietf-idmr-igmp-mib-13.txt” MUST be implemented by DOCSIS 1.1 compliant CMTS and CMs.

### **3.3.14 Requirements for “draft-ietf-ipcdn-bpiplus-mib-02.txt” BPI+ MIB**

“draft-ietf-ipcdn-bpiplus-mib-02.txt” MUST be implemented by DOCSIS 1.1 compliant CMTS and CMs.

### **3.3.15 Requirements for “draft-ietf-xxxx-xxxx-xxxx-00.txt” USB MIB**

“draft-ietf-xxxx-xxxx-xxxx-00.txt” MUST be implemented by DOCSIS 1.1 compliant CMs that support USB.

### **3.3.16 Requirements for “draft-ietf-ipcdn-subscriber-mib-01.txt” Subscriber Management**

## MIB

“draft-ietf-ipcdn-subscriber-mib-01-.txt” MUST be implemented by DOCSIS 1.1 compliant CMTS.

DOCSIS 1.1 compliant CMTS MUST support a minimum number of filter groups; (30) thirty groups of (20) twenty filters each.

### 3.3.17 Requirements for RFC-2786 Diffie-Helman USM Key

RFC-2786 MUST be implemented by DOCSIS 1.1 compliant CMs. It MAY be implemented on the CMTS.

## 3.4 CM Configuration Files, TLV-11 and MIB OIDs/Values

The following sections define the use of CM configuration file TLV-11 elements and the CM rules for translating TLV-11 elements into SNMP PDU (SNMP MIB OID/instance and MIB OID/instance value combinations; also referred to as SNMP varbinds).

This section also defines the CM behaviors, or state transitions, after either pass or fail of the CM configuration process.

For TLV-11 definitions refer to [MCNS 5; Appendix C].

### 3.4.1 CM configuration file TLV-11 element translation (to SNMP PDU)

TLV-11 translation defines the process used by CM to convert CM configuration file information (TLV-11 elements) into SNMP PDU (varbinds). The CM is responsible for translating CM configuration file TLV-11 elements into a single SNMP PDU containing (n) MIB OID/instance and value components (SNMP varbinds). Once a single SNMP PDU is constructed, the CM will process the SNMP PDU and determine CM configuration pass/fail based on the rules for CM configuration file processing, described below. However, if a CM is not physically capable of processing a, potentially large, single CM configuration file generated SNMP PDU, then the CM must still behave as if all MIB OID/instance and value components (SNMP varbinds), from CM configuration file TLV-11 elements, are processed as a single SNMP PDU.

In accordance with [RFC-1905], the single CM configuration file generated SNMP PDU will be treated “as if simultaneous” and the CM must behave consistently, regardless of the order in which TLV-11 elements appear in the CM configuration file, or SNMP PDU. The singular CM configuration file generated SNMP PDU requirement is consistent with SNMP PDU packet behaviors, received from an SNMP manager; SNMP PDU varbind order does not matter, and there is no defined MAX SNMP PDU limit.

The CM configuration file MUST NOT contain duplicate TLV-11 elements (duplicate means SNMP MIB object has either identical OID or OID from the old and new MIB that actually point to the same SNMP MIB object). If duplicate TLV-11 elements are received by the CM, from the CM configuration file, then the CM MUST fail CM configuration.

#### 3.4.1.1 Rules for CreateAndGo and CreateAndWait

The CM MUST support CreateAndGo for row creation.

The CM MAY support CreateAndWait; with the constraint that CM configuration file TLV-11 elements MUST NOT be duplicated (all SNMP MIB OID/instance must be unique). For instance, an SNMP PDU, constructed from CM configuration file TLV-11 elements, which contains an SNMP CreateAndWait value, for a given SNMP MIB OID/instance, MAY NOT also contain an SNMP Active value for the same SNMP MIB OID/instance (and vice versa). A CM configuration file MAY contain a TLV-11 CreateAndWait element if the intended result is to create an SNMP table row which will remain in the SNMP NotReady or SNMP NotInService state until a non-configuration file SNMP PDU is issued, from an SNMP manager, to update the SNMP table row status.

Both SNMP NotReady and SNMP NotInService states are valid table row states after an SNMP CreateAndWait instruction.

### **3.4.2 Ignore CM configuration TLV-11 elements which are not supported by CM**

If any CM configuration file TLV-11 elements translate to SNMP MIB OID's that are not MIB OID elements supported by the CM, then those SNMP varbinds MUST be ignored, and treated as if they had not been present, for the purpose of CM configuration. This means that the CM will ignore SNMP MIB OIDs for other vendor's private MIB's as well as standard MIB elements that the CM does not support.

CMs that do not support SNMP CreateAndWait for a given SNMP MIB table MUST ignore, and treated as if not present, the set of columns associated with the SNMP table row.

If any CM configuration file TLV-11 element(s) are ignored, then the CM MUST report via the CM configured notification mechanism(s), after the CM is registered. The CM notification method MUST be in accordance with the "Standard DOCSIS event" section, defined within this document.

### **3.4.3 CM state after CM configuration file processing success**

CM proceeds to register, and pass data.

### **3.4.4 CM state after CM configuration file processing failure**

If any CM configuration file generated SNMP PDU varbind performs an illegal set operation (illegal, bad, or inconsistent value) to any MIB OID/instance supported by the CM, then processing of the CM configuration file MUST fail. Any CM configuration file generated SNMP PDU varbind set failure MUST cause a CM configuration failure, and the CM MUST NOT proceed with CM registration.

## 4 OSSI for Radio Frequency Interface

### 4.1 Subscriber Account Management Interface Specification

The subscriber account management requirement and policy will be specified in this section by the ECR/ECO/ECN process.

### 4.2 Configuration Management

Configuration management is concerned with initializing, maintaining, adding and updating network components. In a DOCSIS environment, this includes a cable modem and/or CMTS. Unlike performance, fault, and account management, which emphasize network monitoring, configuration management is primarily concerned with network control. Network control, as defined by this interface specification, is concerned with modifying parameters in and causing actions to be taken by the cable modem and/or CMTS. Configuration parameters could include both identifiable physical resources (for example, Ethernet Interface) and logical objects (for example, IP Filter Table).

Modifying the configuration information of a CM and/or CMTS can be categorized as follows:

- Non-operational
- Operational

Non-operational changes occur when a manager issues a modify command to a CM/CMTS, and the change doesn't effect the operating environment. For example, a manager may change contact information, such as the name and address of the person responsible for a CMTS.

Operational changes occur when a manager issues a modify command to a CM/CMTS, and the change affects the underlying resource or environment. For example, a manager may change the docsDevResetNow object from false to true, which in turn will cause the CM to reboot.

To adjust the necessary attribute values, the CM and CMTS MUST support MIB objects as specified in section 3 of this document.

While the network is in operation, configuration management will be responsible for monitoring the configuration and making changes in response to commands via SNMP or in response to other network management functions.

For example, a *performance management function* may detect that response time is degrading due to a high number of uncorrected frames, and may issue a configuration management change to modify the modulation type from 16Qam to QPSK. A *fault management function* may detect and isolate a fault and may issue a configuration management change to bypass the fault.

#### 4.2.1 Version Control

The CM and CMTS SHOULD support software revision and operational parameter configuration interrogation.

The CM MUST (and the CMTS SHOULD) include at least the hardware version, Boot ROM image version and vendor name in the sysDescr object (from [RFC-1907]). To avoid duplication of management information, the CM (and CMTS) SysDescr MUST NOT contain the software version information; however, if the CMTS does not implement the optional docsDevSwCurrentVers MIB object, then the CMTS SysDescr MUST contain the software

version information.

The format of the specific information contained in the sysDescr MUST be as follows:

To report	Format of each field
Hardware Version	HW_REV: <Hardware version>
Vendor Name	VENDOR: <Vendor name>
Boot ROM	BOOTR: <Boot ROM Version>

Each type value pair MUST be separated with a colon and blank space. Each pair is separated by a “;” followed by a blank. For instance, a sysDescr of a CM of vendor X, hardware version 5.2, and Boot ROM version 1.4

MUST appear as following:

any text<<HW\_REV: 5.2; VENDOR: X; BOOTR: 1.4>>any text

The intent of specifying the format of sysObjectID and sysDescr is to define how to report information in a consistent manner so that sysObjectID and sysDescr field information can be programmatically parsed. This format specification does not intend to restrict the vendor’s hardware version numbering policy.

The CM MUST (and the CMTS is optional) implement the docsDevSwCurrentVers object ([RFC 2669]) to report the current software version.

## 4.2.2 System Initialization and Configuration

There are several methods available to configure CM and CMTS including console port, SNMP set, configuration file, and configuration-file-based SNMP encoded object. The CM MUST support system initialization and configuration via configuration file, configuration-file-based SNMP encoded object and SNMP set. The CMTS MUST support system initialization and configuration via telnet connection, console port, and SNMP set. The CM and CMTS (only CMTS that support configuration by configuration file) MUST support any valid configuration file regardless of configuration file size.

## 4.2.3 Secure Software Upgrades

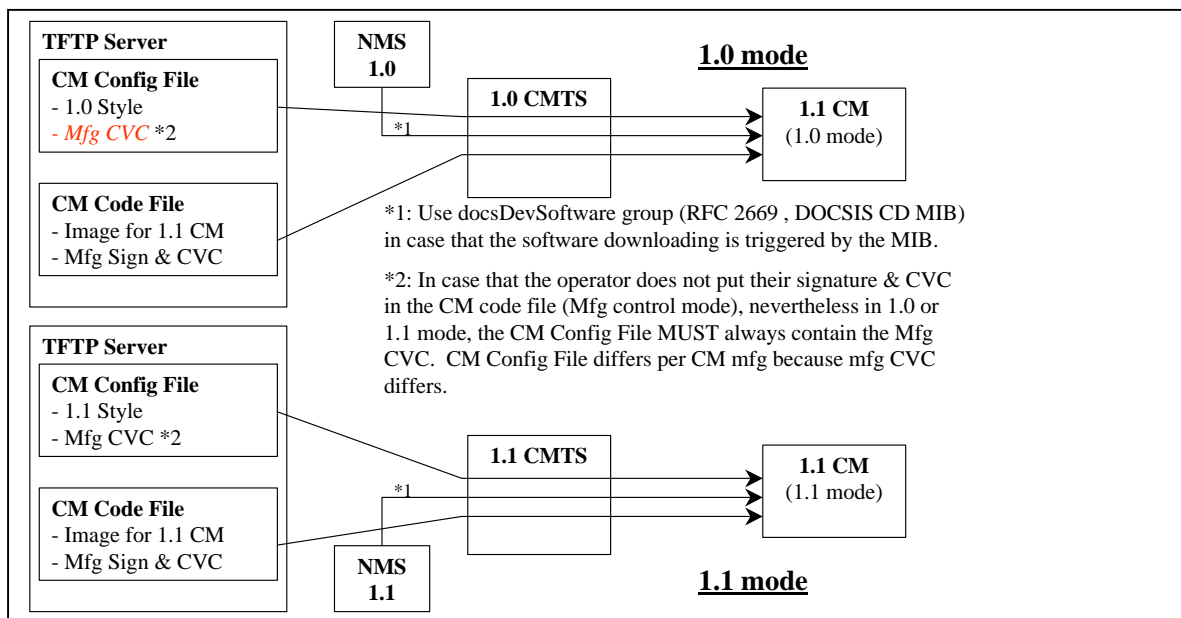
The CM secure software upgrade detail process is documented in the Appendix D of BPI+ specification.

DOCSIS 1.1 CM MUST use secure software upgrade mechanism to perform software upgrade regardless of what DOCSIS CMTS version (1.0 or 1.1) it is connected to. When a 1.1 CM is connected to a 1.1 CMTS, the 1.1 CM operates in DOCSIS 1.1 mode. When a 1.1 CM is connected to a 1.0 CMTS, the 1.1 CM operates in DOCSIS 1.0 mode. This means that a DOCSIS 1.1 CM MUST use secure software upgrade mechanism to perform software upgrade regardless of what mode it operates in (1.0 mode or 1.1 mode).

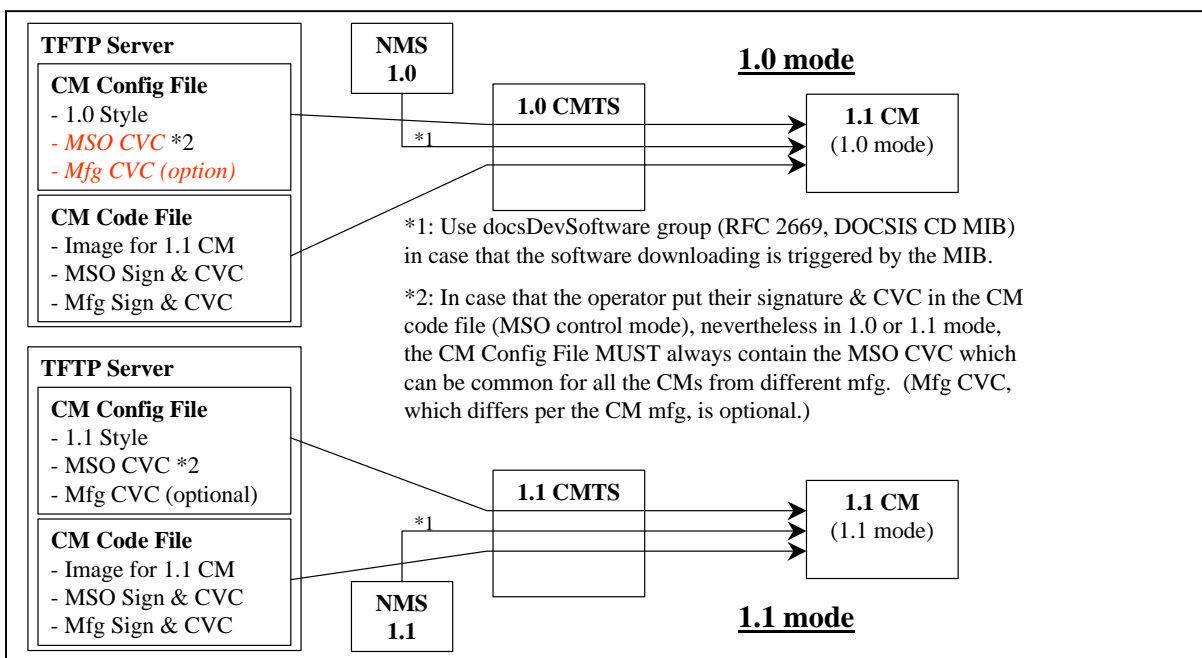


There are two available secure software download schemes including manufacture control scheme and operator control scheme.

1. Manufacture control scheme:



2. Operator control scheme:



Prior to secure software upgrade initialization, CVC information is needed to be initialized at the CM for software upgrade. Depending on the scheme (described above) that the operator chooses to implement, appropriate CVC information **MUST** be include in the configuration file. It is recommended that CVC information always be present in the configuration file so that a device will always have the CVC information initialized and read if the operator decides to use SNMP-initiate upgrade as a method to trigger a secure software upgrade operation. If the operator decides to use configuration-file-initiate upgrade as a method to trigger secure software download, CVC information is needed to be present in the configuration file at the time the modem is rebooted to get the configuration file that will trigger the upgrade only.

There are two methods to trigger secure software download including SNMP-initiated and configuration-file-initiated. Both methods **MUST** be supported by CM and **MAY** be supported by CMTS.

The following describes the SNMP-initiated mechanism. Prior to SNMP-initiate upgrade, a device **MUST** have valid X.509 compliant code verification certificate information. From a network management station:

- Set docsDevSwServer to the address of the TFTP server for software upgrades
- Set docsDevSwFilename to the file pathname of the software upgrade image
- Set docsDevSwAdminStatus to Upgrade-from-mgt.

docsDevSwAdminStatus **MUST** persist across reset/reboots until over-written from an SNMP manager or via the CM configuration file.

The default state of docsDevSwAdminStatus **MUST** be allowProvisioning Upgrade{2} until it is over-written by ignoreProvisioningUpgrade{3} following a successful SNMP initiated software upgrade or otherwise altered by the management station.

docsDevSwOperStatus **MUST** persist across resets to report the outcome of the last software upgrade attempt.

If a CM suffers a loss of power or resets during SNMP-initiated upgrade, the CM **MUST** resume the upgrade without requiring manual intervention. When the CM resumes the upgrade process:

- docsDevSwAdminStatus **MUST** be Upgrade-from-mgt{1}
- docsDevSwFilename **MUST** be the filename of the software image to be upgraded
- docsDevSwServer **MUST** be the address of the TFTP server containing the software upgrade image to be upgraded
- docsDevSwOperStatus **MUST** be inProgress{1}
- docsDevSwCurrentVers **MUST** be the current version of software that is operating on the CM

In case where the CM reaches the maximum number of retries (max retries = 3) resulting from multiple loss of powers or resets during either SNMP-initiated upgrade or configuration-file-initiated upgrade, the CM **MUST** behave as specified in [MCNS5]. In addition, the CM's status **MUST** adhere to the following requirements after it is registered:

- docsDevSwAdminStatus **MUST** be allowProvisioningUpgrade{2}
- docsDevSwFilename **MUST** be the filename of the software that failed the upgrade process.
- docsDevSwServer **MUST** be the address of the TFTP server containing the software that failed the upgrade process
- docsDevSwOperStatus **MUST** be other{5}
- docsDevSwCurrentVer **MUST** be the current version of software that is operating on the CM

If a CM exhausts the required number of TFTP retries by issuing a total of 16 consecutive retries, the CM **MUST** behave as specified in [MCNS5]. Then the CM **MUST** fall back to last known working image and proceed to an operational state and adhere to the following requirements:

- docDevSwAdminStatus MUST be allowProvisioningUpgrade{2}
- docDevSwFilename MUST be the filename of the software that failed the upgrade process
- docsDevSwServer MUST be the address of the TFTP server containing the software that failed the upgrade process
- docsDevSwOperStatus MUST be failed{4}
- docsDevSwCurrentVer MUST be the current version of software that is operating on the CM

After the CM has completed the SNMP-initiated secure software upgrade, the CM MUST behave as specified in [MCNS5] and MUST adhere to the following requirements:

- set its docsDevSwAdminStatus to ignoreProvisioningUpgrade{3}
- set its docsDevOperStatus to completeFromMgt{3}
- reboot

The CM MUST properly use ignoreProvisioningUpgrade status to ignore software upgrade value that may be included in the CM configuration file and become operation with the correct software image. After the CM is registered, it MUST adhere to the following requirements:

- docsDevSwAdminStatus MUST be ignoreProvisioningUpgrade{3}
- docsDevSwFilename MAY be the filename of the software currently operating on the CM
- docsDevSwServer MAY be the address of the TFTP server containing the software that is currently operating on the CM
- docsDevSwOperStatus MUST be completeFromMgt{3}
- docsDevSwCurrentVer MUST be the current version of the software that is operating on the CM

After the CM has completed the configuration-file-initiated secure software upgrade, the CM MUST behave as specified in [MCNS5] and MUST reboot and become operational with the correct software image. After the CM is registered, it MUST adhere to the following requirements:

- docsDevSwAdminStatus MUST be allowProvisioningUpgrade{2}
- docsDevSwFilename MAY be the filename of the software currently operating on the CM
- docsDevSwServer MAY be the address of the TFTP server containing the software that is currently operating on the CM
- docsDevSwOperStatus MUST be completeFromProvisioning{2}
- docsDevSwCurrentVer MUST be the current version of the software that is operating on the CM

In the case where CM successfully downloads (or detects during download) an image that is not intended for the CM device, the CM MUST behave as specified (refer to [MCNS 5], section 10.1 “Downloading Cable Modem Operating Software”):

- DocsDevSwAdminStatus MUST be allowProvisioingUpgrade{2}
- DocsDevSwFilename MUST be the filename of the software that failed the upgrade
- DocsDevSwServer MUST be the address of the TFTP server containing the software that failed the upgrade process
- DocsDevSwOperStatus MUST be other{5}
- DocsDevSwCurrentVer MUST be the current version of software that is operating on the CM

In the case where CM determines that the download image is damaged or corrupted, the CM MUST reject the newly downloaded image. The CM MAY re-attempt to download if the MAX number of TFTP sequence retry has not been reached. If the CM chooses not to retry and the MAX number of TFTP sequence retry has not been reached, the CM MUST fall back to the last known working image and proceed to an operational state, generate appropriate event

notification as specified in Appendix F, and adhere to the following requirements:

- DocsDevSwAdminStausts MUST be allowProvisioningUpgrade{2}
- DocsDevSwFilename MUST be the filename of the software that failed the upgrade
- DocsDevSwServer MUST be the address of the TFTP server containing the software that failed the upgrade process
- DocsDevSwOperStatus MUST be other{5}
- DocsDevSwCurrentVer MUST be the current version of software that is operating on the CM

In the case where CM determines that the image is damaged or corrupted, the CM MUST reject the newly downloaded image. The CM MAY re-attempt to download the new image if the MAX number of TFTP sequence retry has not been reached. On the 16th consecutive failed CM software download attempt, the CM MUST fall back to the last known working image and proceed to an operational state. In this case, the CM is required to send two notifications, one to notify that the MAX TFTP retry limit has been reached, and another to notify that the image is damaged. Immediately after the CM reaches the operational state the CM MUST adhere to the following requirements:

- DocsDevSwAdminStausts MUST be allowProvisioningUpgrade{2}
- DocsDevSwFilename MUST be the filename of the software that failed the upgrade
- DocsDevSwServer MUST be the address of the TFTP server containing the software that failed the upgrade process
- DocsDevSwOperStatus MUST be other{5}
- DocsDevSwCurrentVer MUST be the current version of software that is operating on the CM

### 4.3 Protocol Filters

The CM MUST implement LLC, IP Spoofing, SNMP Access, and IP protocol filters. The LLC protocol filter entries can be used to limit CM forwarding to a restricted set of network-layer protocols (such as IP, IPX, NetBIOS, and AppleTalk). The IP protocol filter entries can be used to restrict upstream or downstream traffic based on source and destination IP addresses, transport-layer protocols (such as TCP, UDP, and ICMP), and source and destination TCP/UDP port numbers.

CM MUST apply filters (or more properly, classifiers) in an order appropriate to the following layering model. Specifically, the inbound MAC (or LLC) layer filters are applied first, then the "special" filters, then the IP layer inbound filters, then the IP layer outbound filters, then any final LLC outbound filters. Note that LLC outbound filters are expected future requirements of the Cable Device MIB.

#### 4.3.1 LLC Filter

The inbound LLC filters are contained in the docsDevFilterLLCTable. These filters are applied to layer-2 frames entering the CM from either the CATV MAC interface{2} and/or any CM CPE interfaces.

The object docsDevFilterLLCUnmatchedAction MUST apply to all interfaces. The default value of the docsDevFilterLLCUnmatchedAction MUST be set to accept.

docsDevFilterLLCUnmatchedAction:

If set to discard(1), any L2 packet that does not match any LLC filters will be discarded, otherwise accepted. If set to accept, any L2 packet that does not match any LLC filters will be accepted, otherwise discarded.

Another way to interpret this is the following:

---

```

action = UnMatchedAction
Iterate through the table
  if there is a match (packet.protocol = row.protocol)
    { reverse the action (accept becomes discard, discard becomes accept)
      apply action to the packet
      terminate the iteration
    }

```

LLC filters MUST apply to in-bound traffic direction only. Traffic generated from CM MUST not be applied to LLC filters (i.e. ARP requests, SNMP responses).

The CM MUST support a minimum of ten LLC protocol filter entries.

### 4.3.2 Special Filter

Special filters are IP spoofing filters and SNMP access filters. IP spoofing filters MUST only be applied to packets entering the modem from CMCI interface(s). SNMP access filters are in effect when the CM is not running in SNMPv3 agent mode and can be applied to both CMCI and CATV interfaces.

According to the interface number section of document, CMCI interface is a generic reference to any current or future form of CM CPE interface port technology.

### 4.3.3 IP Spoofing Filter

CM MUST build a one-to-one relationship (link) between learned CPE MAC and learned IP.

DOCSIS 1.1 compliant CM MUST support anti-spoofing filter function with the following rules:

1.) All DHCP/BOOTP requests go through the anti-spoofing filter regardless of the filter settings in the docsDevCpeTable. In the learning mode, the docsDevCpeTable MUST NOT learn the source IP address of the DHCP/BOOTP packets.

2.) CM MUST populate docsDevCpeTable with IP address from IP packet from CPE and DHCP transaction packets with the following rules:

Assumption: the following rules apply to packets containing MAC addresses that are allowed to be bridged by the CM. CPEIPMAX <> -1.

In the learning mode (docsDevCpeTable is not full):

- Upon receiving a CPE's IP packet, if the source IP is not in the docsDevCpeTable and source MAC address has no link to a source IP in docsDevCpeTable CM MUST populate the source IP address in the docsDevCpeTable. CM MUST also build a link between the source IP address and the source MAC address.
- Upon receiving a CPE's IP packet, if the source IP address is not in the docsDevCpeTable and MAC address has a link to a source IP address in docsDevCpeTable, CM MUST discard the packet.

In the learning mode (docsDevCpeTable is full)

- Upon receiving a CPE's IP packet, CM MUST only accept matching traffic (packet's source IP address matches an entry of docsDevCpeTable. All the unmatched traffic (except mentioned in p.1 ) MUST be discarded.

A DOCSIS 1.1 compliant CM MUST implement the SNMP object docsDevCpeIpMax with a default value of -1.

#### 4.3.3.1 DocsDevCpeIpMax, TLV type-18 (Maximum Number of CPEs) and there relationship with FilterCpeTable

The docsDevCpeIpMax value specifies the MAX number of docsDevCpeTable rows, and the TLV type-18 (Maximum Number of CPEs) value specifies the MAX number of CPE MAC address CM is allowed to bridge/forward. When TLV type-18 value is less than docsDevCpeIpMax value, the TLV type-18 value establishes the MAX number of docsDevCpeTable rows; otherwise, the docsDevCpeIpMax value establishes the MAX number of docsDevCpeTable rows.

Handling of configuration file containing both TLV type-18 value (>1) and docsDevCpeIpMax value (>1):

If docsDevCpeIpMax value is greater than TLV type-18 value, CM MUST limit the number of rows in the docsDevCpeTable to the TLV type-18 value.

Handling of configuration file with docsDevCpeIpMax value but no TLV type-18 value:

**The [MCNS 5] (TLV type-18) requirement states that if TLV type-18 is not specified in the configuration file, the CM MUST default Maximum Number of CPEs to 1 (refer to section C.1.1.7 of MCNS5).**

If TLV type-18 is not supplied in the CM configuration file and docsDevCpeIpMax value is > 1, CM MUST limit the number of row(s) in the docsDevCpeTable to 1.

If TLV type-18 is not supplied in the CM configuration file and docsDevCpeIpMax value is = 0, CM MUST limit the number of row(s) in the docsDevCpeTable to 1.

#### 4.3.4 SNMP Access Filter

The SNMP access filters are applied to SNMP packets entering from any interfaces and destined for the CM. SNMP access filter MUST be applied after IP spoofing filters for the packets entering the CM from the CMCI interface. Since SNMP access filter function is controlled by docsDevNmAccessTable, SNMP access filter is available and applies only when the CM is in SNMPv1 or SNMPv2c mode.

When CM is running in SNMPv3 mode SNMP access is controlled and specified in the User Security Model MIB [RFC-2774].

docsDevNmAccessIP and docsDevNmAccessIpMask :

The device that implement docsDevNmAccessTable MUST apply the following rule in order to determine whether to permit SNMP access from a SrcIpAddr:

The NmAccessIpMask MUST be set to 0.0.0.0 in order to allow any NMS. The default value of the docsDevNmAccessIpMask MUST be set to '0.0.0.0'.

if ((NmAccessIp AND NmAccessIpMask)) == (SrcIpAddr AND NmAccessIpMask))

Permit the access from SrcIpAddr;

else

Do NOT permit the access from SrcIpAddr

Allow any NMS:

NmAccessIP = any IP address

NmAccessIpMask = 0.0.0.0

Allow single NMS:

NmAccessIP = an IP address

NmAccessIpMask = 255.255.255.255

Allow group of IP:

NmAccessIP = IP address of the IP subnet

NmAccessIPMask = Netmask of the subnet

Not allow any IP:

NmAccessIP = 0.0.0.0

NmAccessIPMask = 255.255.255.255

### 4.3.5 IP Filter

The object docsDevFilterIPDefault MUST apply to all interfaces. DOCSIS 1.1 compliant CM MUST support a minimum 16 IP filters.

## 4.4 Fault Management

The goals of fault management are remote monitoring/detection, diagnosis, and correction of problems. Network Management operators rely on the ability to monitor and detect problems(s) (such as ability to trace and identify faults, accept and act on error-detection events), as well as the ability to diagnose and correct problem(s) (such as perform a sequences of diagnostic tests, correct faults, and display/maintain event logs.)

This section defines what MUST be available to support remote monitoring/detection, diagnosis and correction of problems.

### 4.4.1 SNMP Usage

In the DOCSIS environment, the goals of fault management are the remote detection, diagnosis, and correction of network problems. Therefore, the CM MUST support SNMP management traffic across both the CPE and CATV MAC interfaces regardless of the CM's connectivity state. CM SNMP access may be restricted to support policy goals. CM installation personnel can use SNMP queries from a station on the CMCI side to perform on-site CM and diagnostics and fault classification (note that this may require temporary provisioning of the CM from a local DHCP server). Further, future CMCI side customer applications, using SNMP queries, can diagnose simple post-installation problems, avoiding visits from service personnel and minimizing help desk telephone queries.

Standard MIB-II support MUST be implemented to instrument interface status, packet corruption, protocol errors, etc. The transmission MIB for Ethernet-like objects [RFC-2665] MUST be implemented on each cable device (CMTS/CM) Ethernet and Fast Ethernet port. Each cable device (CMTS/CM) MUST implement the ifXTable [RFC-2233] to provide discrimination between broadcast and multicast traffic.

The cable device (CMTS/CM) MUST support managed objects for fault management of the PHY and MAC layers. The RFC-2670 MIB includes variables to track PHY state such as codeword collisions and corruption, signal-to-noise ratios, transmit and receive power levels, propagation delays, micro-reflections, in channel response, and Sync loss. The RFC-2670 MIB also includes variables to track MAC state, such as collisions and excessive retries for requests, immediate data transmits, and initial ranging requests.

For fault management at all layers, the cable device (CMTS/CM) MUST generate replies to SNMP queries (subject to policy filters) for counters and status. The cable device (CMTS/CM) MUST send SNMP traps to one or more trap NMSs (subject to policy), and MUST send SYSLOG events to a SYSLOG server (if a SYSLOG server is defined).

When the cable device (CM) is operating in SNMPv1/v2c mode it MUST support the capability of sending traps as specify by the following MIB object (proposed MIB extension to the docsDevNmAccess table):

DocsDevNmAccessTrapVersion OBJECT-TYPE

```
SYNTAX      INTEGER {
    DisableSNMPv2trap(1),
    EnableSNMPv2trap(2),
}
```

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Specifies the TRAP version that is sent to this NMS. Setting this object to

DisableSNMPv2trap (1) causes the trap in SNMPv1 format to be sent to particular NMS. Setting this object to EnableSNMPv2trap (2) causes the trap in SNMPv2 format be sent to particular NMS"

DEFVAL { Disable SNMPv2trap }

::= { docsDevNmAccessEntry 8 }

Any cable device (CMTS/CM) SHOULD implement the ifTestTable [RFC-2233] for any diagnostic test procedures that can be remotely initiated.

#### 4.4.2 Event Notification

A cable device (CMTS/CM) MUST generate asynchronous events that indicate malfunction situations and notify about important non-fault events. Events could be stored in CMTS/CM device internal event LOG file, in non-volatile memory, get reported to other SNMP entities (as TRAP or INFORM SNMP messages), or be sent as a SYSLOG event message to a pre-defined SYSLOG server. Events MAY also be sent to the cable device (CMTS/CM) console; as a duplicate (identical) message to the optional console destination.

Event notification implemented by a cable device (CMTS/CM) MUST be fully configurable, by priority class; including the ability to disable SNMP Trap, SYSLOG transmission, and local logging. CMTS/CM MUST implement docsDevEvControlTable to control reporting of event classes. The object docsDevEvReporting MUST be implemented as RW for CMTS/CM.



A cable device (CMTS/CM) **MUST** support the following event notification mechanisms (regardless of what SNMP mode the cable device is in):

- local event logging
- SNMP TRAP/INFORM (trap-versions/targets/limiting/throttling)
- SYSLOG (targets/limiting/throttling)
- (Refer to the following sections for event notification implementation details)

When a CM is in SNMPv1/SNMPv2c mode, the CM **MUST** support event notification functions including local event logging, SYSLOG (targets/limiting/throttling) and SNMP TRAP (trap-versions/targets/limiting/throttling) as specified in RFC-2669. When CM is in SNMPv3 mode, CM **MUST** support event notification functions including local event logging, SYSLOG (targets/limiting/throttling) and SNMP TRAP (limiting/throttling) as specified in RFC-2669 and SNMP TRAP targets as specified in RFC-2573.

When a CMTS is in SNMPv1/SNMPv2c mode, the CMTS **MUST** support event notification functions including local event logging, SYSLOG (targets/limiting/throttling) and SNMP TRAP (limiting/throttling) as specified in RFC-2669; however, SNMP TRAP (trap-versions/targets) **MAY** be implemented as specified in RFC-2669, or vendor proprietary MIB. When CMTS is in SNMPv3 mode, CMTS **MUST** support event notification functions including local event logging, SYSLOG (targets/limiting/throttling) and SNMP TRAP (limiting/throttling) as specified in RFC-2669 and SNMP TRAP (targets) as specified in RFC-2573.

#### **4.4.2.1 Local Event Logging**

Event logging provides a mechanism to store critical and error events in non-volatile memory. The event log storage, and access mechanism, **MUST** be implemented in cable device (CMTS/CM) as described below. A DOCSIS 1.1 compliant cable device (CMTS/CM) **MUST** implement docsDevEventTable with additional requirement as specify by the OSSI 1.1

The cable device (CMTS/CM) event log **MUST** be organized as a cyclic buffer with a minimum event depth of ten entries, and **MUST** persist across CMTS/CM cable device re-boot. In the event of a full CMTS/CM event log, new log events **MUST** overwrite the oldest event log entry. The event log table **MUST** be accessible through docsDevEventTable [RFC2669] by cable device (CMTS/CM). Only events of the priority which has Local Log enabled **MUST** be stored in the local event log and appear in the docsDevEventTable. Each entry in the docsDevEventTable contains an event-ID, event time stamp (time the event occurred), event priority level, and event description (in human-readable English format). A network management application could periodically poll the event log table and retrieve event log entries.

Because of the cyclic nature of the local event log, only one event can be associated with one event log entry. This means that docsDevEvLastTime is always equal to docsDevEvFirstTime and docsDevEvCounts **MUST** always be equal to one. This is different from the current definition of the docsDevEventTable in RFC2669, which can't be used as a cyclic buffer, because it allows multiple events of the same type to be stored in one event log entry.

#### **4.4.2.2 Format of Events**

##### **4.4.2.2.1 SNMP TRAP/Inform**

All SNMP TRAPs associated with the events described in this document are defined in the corresponding parts of standard DOCSIS MIB-s (CABLE-DEVICE-MIB, BPI-PLUS-MIB, and DOCS-IF-MIB).

A cable device (CMTS/CM) **MUST** send the following generic SNMP TRAPs, as defined in standard MIB [RFC1907] and [RFC2233]:

- coldStart (warmStart is optional) [RFC-2233]
- linkUp [RFC-2233]
- linkDown [RFC-2233]
- SNMP authentication-Failure [RFC-1907]

Vendor-specific events reportable via SNMP TRAP MUST be described in the vendor private MIBs. The last digit of the MIB OID MUST be the EventId of the event associated with the trap. The event-ID digit is a 32-bit unsigned integer. EventId's from 0 to 2<sup>31</sup>-1 are reserved by DOCSIS. The event-ID (number) is converted from the error codes defined in Appendix J of [MCNS5] (as described in the "SYSLOG Message Format" section).

The EventId from 2<sup>31</sup> to 2<sup>32</sup>-1 will be used as vendor-specific event-ID with the following requirements:

- Bit 31 top bit set on = Vendor Specific Event
- Put the bottom 15 bits of the vendor's SNMP enterprise number in bits 30 to 16
- Bits 15 to 0 can be used by vendor for events (65,535 possible events per vendor)

TRAPS in SNMPv1/v2c mode could be sent either in SNMPv1 or SNMPv2c format. TRAPS in SNMPv3 mode MUST be sent in SNMPv2c format. Cable device (CMTS/CM) MUST support INFORM.

The standard DOCSIS TRAPS and vendor-specific TRAPS in SNMPv1 format MUST have the 'enterprise' field set to the MIB OID of the ROOT part, of the notification section, of the corresponding MIB as defined in (CABLE-DEVICE-MIB, BPI-MIB or the private vendor MIB). The 'trap-specific' field, of SNMPv1 TRAP PDU, MUST have the EventId code of the corresponding event. This way the 'enterprise' field together with an OID of 0 and the 'specific-trap' will form the SNMPv2 trap OID for this trap. For example, for the Cable Device MIB, the enterprise field is {docsDev 2}. For a specific trap with eventID code of 1140850944, the SNMPv2 trap OID would be {docsDev 2.0.1140850944}. This is in accordance with RFC-2576, section 3.1, paragraph 2.

The TRAPS in SNMPv2c format must contain the sysUpTime.0 object and the snmpTrapOID.0 object set to the corresponding TRAP MIB OID.

Traps in both SNMPv1 and SNMPv2c format must include the object docsDevEvText, set to the textual description of the event, which MUST be identical to the text that is included in SYSLOG message and stored in the local event log file. Also, Traps MUST include all objects that are defined in the particular SNMP trap definition.

#### **4.4.2.2.2 SYSLOG Message Format**

CM's Syslog message MUST be sent in the following format:

`<level>CABLEMODEM[<vendor>]: <eventId> text`

Where:

*Level* - ASCII presentation of the event priority, enclosed in angle brackets, which is constructed as OR of the default Facility (128) and event priority (0-7). The resulted level has the range between 128 and 135.

CMTS's Syslog message MUST be sent in the following format:

`<level>CMTS[<vendor>]: <eventId> text`

Where:

*Level* - ASCII presentation of the event priority, enclosed in angle brackets, which is constructed as OR of the default Facility (128) and event priority (0-7). The resulted level has the range between 128 and 135

*vendor* - Vendor name for the vendor-specific SYSLOG messages or DOCSIS for the standard DOCSIS messages.

*EventId* - ASCII presentation of the INTEGER number in HEX format, enclosed in angle brackets, which uniquely identifies the type of event. This number **MUST** be the same number that is stored in docsDevEvId object in docsDevEventTable and also is associated with SNMP TRAP in the “SNMP TRAP/Inform” section. For the standard DOCSIS events specified in [SP-RFiv1.1] this number is converted from the error codes [SP-RFiv1.1 Appendix J] using the following rules:

- The first byte (MSB) of the EventID is ASCII code for the letter in the Error code from Appendix J.
- Next 2 bytes of the EventID represent 2 or 3 digits before the dot in the Error code.
- The last byte is the number after the dot in the Error code.

**For example, event D04.2 has the number (in HEX) 0x44000402**

**Event I114.1 has the number 0x49007201**

*text* - for the standard DOCSIS messages this string **MUST** have the textual description as defined in [SP-RFiv1.1 Appendix J]. For the vendor-specific messages it could be any ASCII single-line string that describes event.

The example of the syslog event for the event D04.2

"Time of the day received in invalid format":

<132>CABLEMODEM[DOCSIS]: <44000402> Time of Day Response but invalid data/format.

The number 44000402 in the given example is the number assigned by DOCSIS to this particular event.

#### 4.4.2.3 Standard DOCSIS Events for CM

The DOCSIS cable device MIB [RFC2669] document defines 8 different priority levels and the corresponding reporting mechanism for each level. The standard DOCSIS events specified in this document utilizes the subset of these priority levels.

Emergency event (priority 1)

Reserved for vendor-specific ‘fatal’ hardware or software errors that prevents normal system operation and causes reporting system to reboot.

**Every vendor may define there own set of emergency events. The examples of such events could be ‘no memory buffers available’, ‘memory test failure’ etc. (Such basic cross-vendor type events should be included in the DOCSIS 1.1 “Events for Notification” Appendix F so that vendors do not define many overlapping EventId’s in vendor-private scope)**

Alert event (priority 2)

A serious failure, which causes reporting system to reboot but it is not caused by h/w or s/w malfunctioning. After recovering from the critical event system **MUST** send the cold/warm start notification. Alert event could not be reported as a Trap or SYSLOG message and **MUST** be stored in the internal log file. The code of this event **MUST** be saved in non-volatile memory and reported later through docsIfCmStatusCode SNMP variable [RFC2670].

Critical event (priority 3)

A serious failure that requires attention and prevents the device from transmitting data but could be recovered without rebooting the system. After recovering from the error event Cable Modem Device **MUST** send the Link Up notification. Critical events could not be reported as a Trap or SYSLOG message and **MUST** be stored in the internal log file. The code of this event **MUST** be reported later through docsIfCmStatusCode SNMP variable

[RFC2670]. The examples of such events could be configuration file problems detected by the modem or inability to get IP address from DHCP.

#### Error event (priority 4)

A failure occurred that could interrupt the normal data flow but does not cause modem to re-register. Error events could be reported in real time by using TRAP or SYSLOG mechanism.

#### Warning event (priority 5)

A failure occurred that could interrupt the normal data flow but does not cause modem to re-register. 'Warning' level is assigned to events both modem and CMTS have information about. So to prevent sending same event both from the CM and CMTS, trap and Syslog reporting mechanism is disabled by default for this level.

#### Notice event (priority 6)

The event of importance which is not a failure and could be reported in real time by using TRAP or SYSLOG mechanism. The examples of the NOTICE events are 'Cold Start', 'Warm Start', 'Link Up' and 'SW upgrade successful'.

#### Informational event (priority 7)

The not-important event, which is not failure, but could be helpful for tracing the normal modem operation. By default these events are not saved into the local event log and no Syslog/trap is sent.

#### Debug event (priority 8)

Reserved for vendor-specific non-critical events

The priority associated with the event is hard-coded and can't be changed. The reporting mechanism for each priority could be changed from the default reporting mechanism (Table 1) by using docsDevEvReporting object in cable device MIB [RFC2669].

Event Priority	Local Log	Trap	Syslog	Note
1 Emergency	Yes	No	No	Vendor-spec.
2 Alert	Yes	No	No	DOCSIS
3 Critical	Yes	No	No	DOCSIS
4 Error	Yes	Yes	Yes	DOCSIS
5 Warning	Yes	No	No	DOCSIS
6 notice	Yes	Yes	Yes	DOCSIS
7 informational	No	No	No	DOCSIS/vend.
8 debug	No	No	No	Vendor-spec.

Table 1 Event priorities for the Cable Modem Device

DOCSIS 1.1 compliant CM MUST generate event notification based on events specified in Appendix F.

#### 4.4.2.4 Standard DOCSIS Events for CMTS

CMTS uses the same levels of the event priorities as a CM; however, the severity definition of the events is different. Events with the severity level of Warning and less specify problems that could affect individual user (for example, individual CM registration problem).

Severity level of ‘Error’ indicates problems with a group of CMs (for example CMs that share same upstream channel).

Severity level of ‘Critical’ indicates problem that affects whole cable system operation, but is not a faulty condition of CMTS device. In all these cases CMTS MUST be able to send SYSLOG event and (or) SNMP TRAP to the NMS.

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

Event Priority	Local Log	Trap	Syslog	Note
1 Emergency	Yes	No	No	Vendor-spec.
2 Alert	Yes	No	No	Vendor-spec.
3 Critical	Yes	Yes	Yes	DOCSIS
4 Error	Yes	Yes	Yes	DOCSIS
5 Warning	No	Yes	Yes	DOCSIS
6 notice	No	Yes	Yes	DOCSIS
7 informational	No	No	No	DOCSIS/vend.
8 debug	No	No	No	Vendor-spec.

Table 2 Default Event priorities for the CMTS Device

DOCSIS 1.1 compliant CMTS MUST generate event notification based on events specified in Appendix F.

#### 4.4.3 Trap and Syslog Throttling, Trap and Syslog Limiting

DOCSIS 1.1 compliant cable device (CMTS/CM) MUST support SNMP TRAP/INFORM and SYSLOG throttling and limiting as described in RFC-2669, regardless of SNMP mode (v1/v2c/v3).

#### 4.4.4 Non-SNMP Fault Management Protocols

The OSS can use a variety of tools and techniques to examine faults at multiple layers. For the IP layer, useful non-SNMP based tools include ping (ICMP Echo and Echo Reply), traceroute (UDP and various ICMP Destination Unreachable flavors). Pings to a CM from its CMCI side MUST be supported to enable local connectivity testing from a customer’s PC to the modem. The CM and CMTS MUST support IP end-station generation of ICMP error messages and processing of all ICMP messages.

### 4.5 Performance Management

At the CATV MAC and PHY layers, performance management focuses on the monitoring of the effectiveness of cable plant segmentation and rates of upstream traffic and collisions. Instrumentation is provided in the form of the standard interface statistics [RFC-2233], as well as the docsifCmtsServiceTable and docsifCmServiceTable entries. It is not anticipated that the CMTS upstream bandwidth allocation function will require active network management

intervention and tuning.

At the LLC layer, the performance management focus is on bridge traffic management. The CM and CMTS (if the CMTS implements transparent bridging) MUST implement the Bridge MIB RFC-1493, including the dot1dBase and dot1dTp groups. The CM and CMTS MUST implement a managed object that controls whether the 802.1d spanning tree protocol (STP) is run and topology update messages are generated; STP is unnecessary in hierarchical, loop-free topologies. If the STP is enabled for the CM/CMTS, then the CM/CMTS MUST implement the dot1dStp group. These MIB groups' objects allow the NMS to detect when bridge forwarding tables are full, and enable the NMS to modify aging timers.

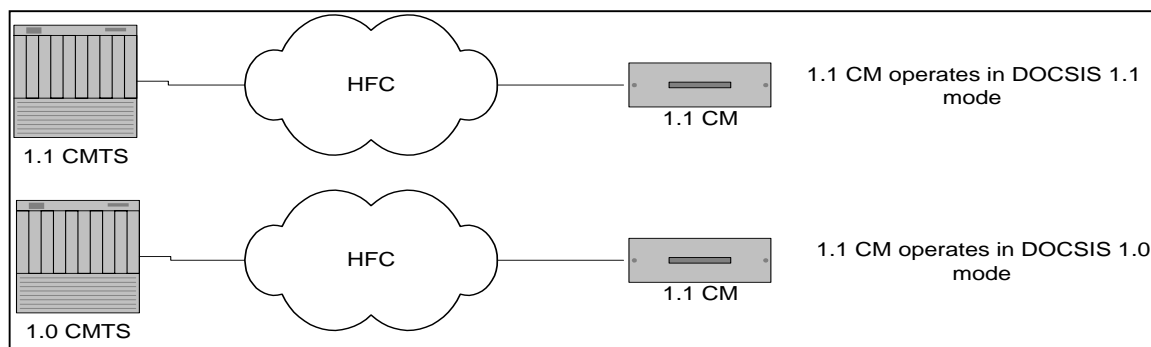
A final performance concern is the ability to diagnose unidirectional loss. Both the CM and CMTS MUST implement the MIB-2 [RFC-2233] Interfaces group. When there exists more than one upstream or downstream channel, the CM/CMTS MUST implement an instance of IfEntry for each channel. The ifStack group [RFC-2233] MUST be used to define the relationships among the CATV MAC interfaces and their channels.

#### **4.5.1 Additional MIB Implementation Requirements**

To support performance monitoring and data collection for capacity, fault, and performance management, CM and CMTS MUST support MIB objects with:

- Accurate in measurement
- Counter properly working (i.e. counter roll over at maximum)
- Correct counter capacity
- Counter reset properly
- Update rate of 1 second

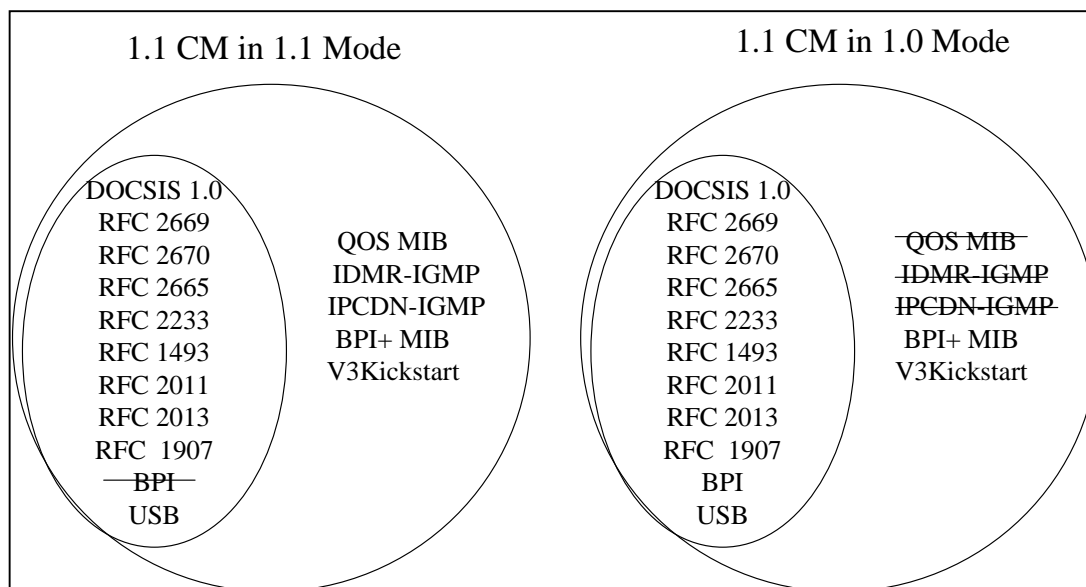
## 4.6 Coexistence



**Figure 3. Coexistent (DOCSIS 1.0 mode VS DOCSIS 1.1 mode)**

When DOCSIS 1.1 compliant CM is connected to 1.1 CMTS, it operates in 1.1 mode. When DOCSIS 1.1 compliant CM is connected to 1.0 CMTS, it operates in 1.0 mode. Refer to [MCNS5] and BPI+ specification for more detail description of what features are available when DOCSIS 1.1 compliant CM is operating in different modes.

### 4.6.1 Coexistence and MIBs



**Figure 4. CM DOCSIS Mode and MIBs Requirement**

**4.6.1.1.1 Requirements for 1.1 CM operating in 1.1 mode**

When DOCSIS 1.1 compliant CM operates in 1.1 mode, it MUST support the following MIBs:

- RFC 2669
- RFC 2670
- RFC 2665
- RFC 1493
- RFC 2011
- RFC 2013
- USB MIB
- QOS MIB
- IDMR-IGMP
- IPCDN-IGMP
- BPI+ MIB
- V3Kickstart [RFC-2786] (When CM is in SNMPv1/v2c mode, CM MUST respond with “NoSuchName” for all the request to tables and objects in V3Kickstart)

When DOCSIS 1.1 compliant CM operates in 1.1 mode, it MUST NOT support the following MIB(s):

- BPI MIB

BPI MIB MUST not be available for any access from SNMP manager. DOCSIS 1.1 compliant CM MUST respond with “NoSuchName” for all requests to tables and objects in BPI MIB.

**4.6.1.1.2 Requirements for 1.1 CM operating in 1.0 mode**

When DOCSIS 1.1 compliant CM operates in 1.0 mode, it MUST support the following MIBs:

- RFC 2669
- RFC 2670
- RFC 2665
- RFC 1493
- RFC 2011
- RFC 2013
- USB MIB
- BPI MIB
- BPI+ MIB. Only part of the BPI+ MIB MUST be supported Additional
- V3Kickstart [RFC-2786] (When CM is in SNMPv1/v2c mode, CM MUST respond with “NoSuchName” for all the request to tables and objects in V3Kickstart)

When DOCSIS 1.1 compliant CM operates in 1.0 mode, it MUST NOT support the following MIB(s):

- QOS MIB
- IDMR-IGMP
- IPCDN-IGMP



- BPI+ (part of the BPI+ MIB MUST be still be supported to enable secure software download. Detail requirement is specified in the BPI+ section of this document.

QOS MIB, IDMR-IGMP MIB, BPI+ MIB, and IPCDN-IGMP MIB MUST not be available for any access from SNMP manager. DOCSIS 1.1 compliant CM MUST respond with “NoSuchName” for all requests to tables and objects in QOS MIB, IDMR-IGMP MIB, IPCDN-IGMP MIB and BPI+ MIB.

#### **4.6.2 Coexistence and SNMP**

DOCSIS 1.1 compliant CM MUST support SNMPv3 and SNMPv1/v2c functionality as specified in Section 2 regardless of what mode (DOCSIS 1.0 or DOCSIS 1.1) CM operates in.

## **5 OSS for BPI+**

The X.509 Digital Certificates management policy and the requirements related to the management of those Digital Certificates will be specified in this section by the ECR/ECO/ECN process.

## 6 OSSI for CMCI

This section defines the operational mechanisms needed to support the transmission of data over cable services between a cable modem and the customer premise equipment. More specifically, this section will outline the following:

- SNMP access via CMCI
- Console Access
- CM diagnostic capabilities
- Protocol Filtering
- Required MIBs

Currently, the CMCI is categorized as internal, external, and CPE Controlled cable modem functional reference models. The external cable modems MAY have either an Ethernet 10BASE-T or Universal Serial Bus (USB) CMCI interface or both. If both interfaces are present on a CM, they MAY be active at the same time.

The internal cable modems MUST utilize the Peripheral Component Interface (PCI) bus for transparent bi-directional IP traffic forwarding. The PCI interface MUST be defined and accessible from an SNMP manager for both operational and security purposes.

The CPE Controlled Cable modems (CCCM) CMCI MAY be either a Peripheral Component Interface (PCI) or Universal Serial Bus (USB) interface. If PCI is utilized, the interface MUST be defined and accessible from an SNMP manager for both operational and security purposes.

### 6.1 SNMP Access via CMCI

A CM MAY be accessible from the Customer Premise Equipment (CPE) utilizing the SNMP protocol. The IP address utilized for CM CPE access MUST originate with vendor and MUST be totally isolated from the IP address assigned by the operator's provisioning server.

A CM device providing CPE SNMP access, prior to completing the CMTS registration process, MUST operate with the following limitations:

- The CM CPE interface SNMP agent MUST support only READ-ONLY access until after the docsDevNmAccess table is set from the CM configuration file (via TFTP Server)
- The CM SNMP agent MUST operate in SNMPv1/v2c mode
- Once the CM has configured the docsDevNmAccess table, via CM configuration file (TFTP Server), the CM CPE interface SNMP agent MUST restrict SNMP access based on the entries specified in the docsDevNmAccess table
- The CM CPE interface SNMP agent MUST prohibit all SNMP access during the SNMPv3 initialization process

### 6.2 Console Access

An external cable modem MUST NOT allow access to the CM functions via a console port. For this specification, a console port is defined as a communication path, either hardware or software that allows a user to issue commands to modify the configuration or operational status of the CM. Access to the external CM MUST only be allowed using DOCSIS 1.1 defined RF interfaces and operator-controlled SNMP access via the CMCI.

### 6.3 CM Diagnostic Capabilities

The cable modem MAY have read-only diagnostic interfaces for debugging and troubleshooting purposes. The read-only diagnostic interface MUST NOT display any network addressing or operational information.

### 6.4 Protocol Filtering

The CM MUST be capable of filtering all broadcast traffic from the host CPE, with the exception of DHCP and ARP packets. This filtering function must adhere to section 4.3 (Protocol Filters) of this document. All ICMP type packets MUST be forwarded from the CMCI interface to the RF upstream interface. The CMCI MUST also adhere to the data forwarding rules defined in [MCNS 5].

### 6.5 Management Information Base (MIB) Requirements

All Cable Modems MUST implement the MIBs detailed in section 3 (Management Information Bases) of this specification, with the following exceptions:

- An external CM with only USB interface(s), MUST NOT implement RFC-2665: Ethernet Interface MIB.
- An external CM with only USB interface(s), MUST implement the IETF Proposed Standard RFC version of USB MIB.
- An internal CM MAY implement RFC-2665: Ethernet Interface MIB.

## Appendix A. Detailed MIB Requirements

NOTE:

D - Deprecated

M - Mandatory

N-Acc - Not accessible

NA - Not Applicable

N-Sup - MUST not support

O - Optional

Ob - Obsolete

RC - Read-Create

RO - Read-Only

RW - Read-Write

RC/RO - Read-Create or Read-Only

RW/RO - Read-Write or Read-Only

General rules:

D - Deprecated – It is optional. That is, 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., no such object for SNMPv2c)

M - Mandatory – The object **MUST** be implemented correctly according to the MIB definition.

N-Acc - Not Accessible – The object is not accessible and is usually an index in a table.

NA - Not Applicable – Not applicable to the device.

N-Sup - MUST Not Support – Device **MUST NOT** support the object. That is, an agent **MUST NOT** instantiate such object and **MUST** respond with the appropriate error/exception condition. (e.g., no such object for SNMPv2c)

O - Optional – 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., no such object for SNMPv2c)

**Ob - Obsolete** – It is optional. Though in SNMP convention, obsolete objects should not be implemented, DOCSIS 1.1 OSSI lets vendors choose whether or not to support the obsolete object. That is, a vendor can choose to implement or not implement the obsolete 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, SNMP agent **MUST NOT** instantiate such object and **MUST** respond with the appropriate error/exception condition. (e.g., no such object for SNMPv2c)

**RC – Read-Create** – The access of the object **MUST** be implemented as Read-Create.

**RO – Read-Only** – The access of the object **MUST** be implemented as Read-Only.

**RW – Read-Write** – The access of the object **MUST** be implemented as Read-Write.

**RC/RO – Read-Create or Read-Only** – The access of the object **MUST** be implemented as either Read-Create or Read-Only as described in the MIB definition.

**RW/RO – Read-Write or Read-Only** – The access of the object **MUST** be implemented as either Read-Write or Read-Only as described in the MIB definition.

<b>DOCS-IF-MIB (RFC 2670)</b>				
<b>docslfDownstreamChannelTable</b>				
<b>Object</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docslfDownChannelId	M	RO	M	RO
docslfDownChannelFrequency	M	RO	M	RW/RO
docslfDownChannelWidth	M	RO	M	RW/RO
docslfDownChannelModulation	M	RO	M	RW
docslfDownChannelInterleave	M	RO	M	RW
docslfDownChannelPower	M	RO	M	RW/RO
<b>docslfUpstreamChannelTable</b>				
<b>Object</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docslfUpChannelId	M	RO	M	RO
docslfUpChannelFrequency	M	RO	M	RW
docslfUpChannelWidth	M	RO	M	RW

docslfUpChannelModulationProfile			M	RO	M	RW
docslfUpChannelSlotSize			M	RO	M	RW/RO
docslfUpChannelTxTimingOffset			M	RO	M	RO
docslfUpChannelRangingBackoffStart			M	RO	M	RW
docslfUpChannelRangingBackoffEnd			M	RO	M	RW
docslfUpChannelTxBackoffStart			M	RO	M	RW
docslfUpChannelTxBackoffEnd			M	RO	M	RW
docslfQosProfileTable						
Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docslfQosProfileIndex	M	N-Acc	O	N-Acc	O	N-Acc
docslfQosProfPriority	M	RO	O	RO	O	RC/RO
docslfQosProfMaxUpBandwidth	M	RO	O	RO	O	RC/RO
docslfQosProfGuarUpBandwidth	M	RO	O	RO	O	RC/RO
docslfQosProfMaxDownBandwidth	M	RO	O	RO	O	RC/RO
docslfQosProfMaxTxBurst	M	RO	O	RO	O	RC/RO
docslfQosProfBaselinePrivacy	M	RO	O	RO	O	RC/RO
docslfQosProfStatus	M	RO	O	RO	O	RC/RO
docslfSignalQualityTable						
Object			CM	Access	CMTS	Access
docslfSigQIncludesContention			M	RO	M	RO
docslfSigQUnerroreds			M	RO	M	RO
docslfSigQCorrecteds			M	RO	M	RO
docslfSigQUncorrectables			M	RO	M	RO
docslfSigQSignalNoise			M	RO	M	RO
docslfSigQMicroreflections			M	RO	M	RO
docslfSigQEqualizationData			M	RO	M	RO
docslfCmMacTable						
Object			CM	Access	CMTS	Access
docslfCmCmtsAddress			M	RO	NA	NA
docslfCmCapabilities			M	RO	NA	NA
docslfCmRangingRespTimeout			Ob	N-Sup	NA	NA
docslfCmRangingTimeout			M	RW	NA	NA
docslfCmStatusTable						
Object			CM	Access	CMTS	Access
docslfCmStatusValue			M	RO	NA	NA
docslfCmStatusCode			M	RO	NA	NA
docslfCmStatusTxPower			M	RO	NA	NA
docslfCmStatusResets			M	RO	NA	NA
docslfCmStatusLostSyncs			M	RO	NA	NA
docslfCmStatusInvalidMaps			M	RO	NA	NA

docslfCmStatusInvalidUcds	M	RO	NA	NA
docslfCmStatusInvalidRangingResponses	M	RO	NA	NA
docslfCmStatusInvalidRegistrationResponses	M	RO	NA	NA
docslfCmStatusT1Timeouts	M	RO	NA	NA
docslfCmStatusT2Timeouts	M	RO	NA	NA
docslfCmStatusT3Timeouts	M	RO	NA	NA
docslfCmStatusT4Timeouts	M	RO	NA	NA
docslfCmStatusRangingAborted	M	RO	NA	NA
<b>docslfCmServiceTable</b>				
<b>Object</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docslfCmServiceId	M	N-Acc	NA	NA
docslfCmServiceQosProfile	M	RO	NA	NA
docslfCmServiceTxSlotsImmed	M	RO	NA	NA
docslfCmServiceTxSlotsDed	M	RO	NA	NA
docslfCmServiceTxRetries	M	RO	NA	NA
docslfCmServiceTxExceededs	M	RO	NA	NA
docslfCmServiceRqRetries	M	RO	NA	NA
docslfCmServiceRqExceededs	M	RO	NA	NA
<b>docslfCmtsMacTable</b>				
<b>Object</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docslfCmtsCapabilities	NA	NA	M	RO
docslfCmtsSyncInterval	NA	NA	M	RW/RO
docslfCmtsUcdInterval	NA	NA	M	RW/RO
docslfCmtsMaxServiceIds	NA	NA	M	RO
docslfCmtsInsertionInterval	NA	NA	Ob	N-Sup
docslfCmtsInvitedRangingAttempts	NA	NA	M	RW/RO
docslfCmtsInsertInterval	NA	NA	M	RW/RO
<b>docslfCmtsStatusTable</b>				
<b>Object</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docslfCmtsStatusInvalidRangeReqs	NA	NA	M	RO
docslfCmtsStatusRangingAborted	NA	NA	M	RO
docslfCmtsStatusInvalidRegReqs	NA	NA	M	RO
docslfCmtsStatusFailedRegReqs	NA	NA	M	RO
docslfCmtsStatusInvalidDataReqs	NA	NA	M	RO
docslfCmtsStatusT5Timeouts	NA	NA	M	RO
<b>docslfCmtsCmStatusTable</b>				
<b>Object</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docslfCmtsCmStatusIndex	NA	NA	M	N-Acc
docslfCmtsCmStatusMacAddress	NA	NA	M	RO
docslfCmtsCmStatusIpAddress	NA	NA	M	RO
docslfCmtsCmStatusDownChannelIfIndex	NA	NA	M	RO
docslfCmtsCmStatusUpChannelIfIndex	NA	NA	M	RO



docsIfCmtsCmStatusRxPower	NA	NA	M	RO
docsIfCmtsCmStatusTimingOffset	NA	NA	M	RO
docsIfCmtsCmStatusEqualizationData	NA	NA	M	RO
docsIfCmtsCmStatusValue	NA	NA	M	RO
docsIfCmtsCmStatusUnerrored	NA	NA	M	RO
docsIfCmtsCmStatusCorrected	NA	NA	M	RO
docsIfCmtsCmStatusUncorrectables	NA	NA	M	RO
docsIfCmtsCmStatusSignalNoise	NA	NA	M	RO
docsIfCmtsCmStatusMicroreflections	NA	NA	M	RO
<b>docsIfCmtsServiceTable</b>				
<b>Object</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsIfCmtsServiceId	NA	NA	M	N-Acc
docsIfCmtsServiceCmStatusIndex	NA	NA	M	RO
DocsIfCmtsServiceAdminStatus	NA	NA	M	RW/RO
docsIfCmtsServiceQosProfile	NA	NA	M	RO
docsIfCmtsServiceCreateTime	NA	NA	M	RO
docsIfCmtsServiceInOctets	NA	NA	M	RO
docsIfCmtsServiceInPackets	NA	NA	M	RO
<b>docsIfCmtsModulationTable</b>				
<b>Object</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsIfCmtsModIndex	NA	NA	M	N-Acc
docsIfCmtsModIntervalUsageCode	NA	NA	M	N-Acc
docsIfCmtsModControl	NA	NA	M	RC
docsIfCmtsModType	NA	NA	M	RC
docsIfCmtsModPreambleLen	NA	NA	M	RC
docsIfCmtsModDifferentialEncoding	NA	NA	M	RC
docsIfCmtsModFECErrorCorrection	NA	NA	M	RC
docsIfCmtsModFECCodewordLength	NA	NA	M	RC
docsIfCmtsModScramblerSeed	NA	NA	M	RC
docsIfCmtsModMaxBurstSize	NA	NA	M	RC
docsIfCmtsModGuardTimeSize	NA	NA	M	RO
docsIfCmtsModLastCodewordShortened	NA	NA	M	RC
docsIfCmtsModScrambler	NA	NA	M	RC
<b>Object</b>				
<b>docsIfCmtsQosProfilePermissions</b>	NA	NA	M	RW /RO
<b>DocsIfCmtsMacToCmTable</b>				
<b>Object</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsIfCmtsCmMac	NA	NA	M	N-Acc
docsIfCmtsCmPtr	NA	NA	M	RO

<b>IF-MIB (RFC 2233)</b>				
<b>Object</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
ifNumber	M	RO	M	RO
IfTableLastChange	M	RO	M	RO
<b>ifTable</b>				
<b>Object</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
ifIndex	M	RO	M	RO
ifDescr	M	RO	M	RO
ifType	M	RO	M	RO
ifMtu	M	RO	M	RO
ifSpeed	M	RO	M	RO
ifPhysAddress	M	RO	M	RO
ifAdminStatus	M	RW	M	RW
ifOperStatus	M	RO	M	RO
ifLastChange	M	RO	M	RO
ifInOctets	M	RO	M	RO
ifInUcastPkts	M	RO	M	RO
ifInNUcastPkts	D	RO	D	RO
ifInDiscards	M	RO	M	RO
ifInErrors	M	RO	M	RO
ifInUnknownProtos	M	RO	M	RO
ifOutOctets	M	RO	M	RO
ifOutUcastPkts	M	RO	M	RO
ifOutNUcastPkts	D	RO	D	RO
ifOutDiscards	M	RO	M	RO
ifOutErrors	M	RO	M	RO
ifOutQLen	D	RO	D	RO
ifSpecific	D	RO	D	RO
<b>ifXTable</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
ifName	M	RO	M	RO
ifInMulticastPkts	M	RO	M	RO
ifInBroadcastPkts	M	RO	M	RO
ifOutMulticastPkts	M	RO	M	RO
ifOutBroadcastPkts	M	RO	M	RO
ifHCInOctets	O	RO	O	RO
ifHCInUcastPkts	O	RO	O	RO
ifHCInMulticastPkts	O	RO	O	RO
ifHCInBroadcastPkts	O	RO	O	RO
ifHCOctets	O	RO	O	RO

ifHCOutUcastPkts	O	RO	O	RO
ifHCOutMulticastPkts	O	RO	O	RO
ifHCOutBroadcastPkts	O	RO	O	RO
ifLinkUpDownTrapEnable	M	RW	M	RW
ifHighSpeed	M	RO	M	RO
ifPromiscuousMode	M	RW/RO	M	RW/RO
ifConnectorPresent	M	RO	M	RO
ifAlias	M	RW/RO	M	RW/RO
ifCounterDiscontinuityTime	M	RO	M	RO
<b>ifStackTable</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
ifStackHigherLayer	M	N-Acc	M	N-Acc
ifStackLowerLayer	M	N-Acc	M	N-Acc
ifStackStatus	M	RC/RO	M	RC/RO
<b>Object</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
ifStackLastChange	O	N-Acc	O	N-Acc
<b>ifRcvAddressTable</b>				
<b>Object</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
ifRcvAddressAddress	O	N-Acc	O	N-Acc
ifRcvAddressStatus	O	RC	O	RC
IfRcvAddressType	O	RC	O	RC
<b>6.5.1.1 Notification</b>				
linkUp	M		M	
linkDown	M		M	
<b>ifTestTable</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
ifTestId	O	RW	O	RW
ifTestStatus	O	RW	O	RW
ifTestType	O	RW	O	RW
ifTestResult	O	RO	O	RO
ifTestCode	O	RO	O	RO
ifTestOwner	O	RW	O	RW
<b>BRIDGE-MIB (RFC 1493)</b>				
NOTE: Implementation of BRIDGE MIB is required ONLY if device is a bridging device				

<b>dot1dBase group</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
dot1dBaseBridgeAddress	M	RO	M	RO
dot1dBaseNumPorts	M	RO	M	RO
dot1dBaseType	M	RO	M	RO
<b>dot1dBasePortTable</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
dot1dBasePort	M	RO	M	RO
dot1dBasePortIfIndex	M	RO	M	RO
dot1dBasePortCircuit	M	RO	M	RO
dot1dBasePortDelayExceededDiscards	M	RO	M	RO
dot1dBasePortMtuExceededDiscards	M	RO	M	RO
<b>dot1dStp group</b>				
NOTE: This group is required ONLY if STP is implemented				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
dot1dStpProtocolSpecification	M	RO	M	RO
dot1dStpPriority	M	RW	M	RW
dot1dStpTimeSinceTopologyChange	M	RO	M	RO
dot1dStpTopChanges	M	RO	M	RO
dot1dStpDesignatedRoot	M	RO	M	RO
dot1dStpRootCost	M	RO	M	RO
dot1dStpRootPort	M	RO	M	RO
dot1dStpMaxAge	M	RO	M	RO
dot1dStpHelloTime	M	RO	M	RO
dot1dStpHoldTime	M	RO	M	RO
dot1dStpForwardDelay	M	RO	M	RO
dot1dStpBridgeMaxAge	M	RW	M	RW
dot1dStpBridgeHelloTime	M	RW	M	RW
dot1dStpBridgeForwardDelay	M	RW	M	RW
<b>dot1dStpPortTable</b>				
NOTE: This table is required ONLY if STP is implemented				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
dot1dStpPort	M	RO	M	RO
dot1dStpPortPriority	M	RW	M	RW
dot1dStpPortState	M	RO	M	RO
dot1dStpPortEnable	M	RW	M	RW
dot1dStpPortPathCost	M	RW	M	RW
dot1dStpPortDesignatedRoot	M	RO	M	RO
dot1dStpPortDesignatedCost	M	RO	M	RO
dot1dStpPortDesignatedBridge	M	RO	M	RO
dot1dStpPortDesignatedPort	M	RO	M	RO
dot1dStpPortForwardTransitions	M	RO	M	RO

<b>dot1dTp group</b>				
<b>Note: This group is required ONLY if transparent bridging is implemented.</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
dot1dTpLearnedEntryDiscards	M	RO	M	RO
dot1dTpAgingTime	M	RW	M	RW
<b>dot1dTpFdbTable</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
dot1dTpFdbAddress	M	RO	M	RO
dot1dTpFdbPort	M	RO	M	RO
dot1dTpFdbStatus	M	RO	M	RO
<b>dot1dTpPortTable</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
dot1dTpPort	M	RO	M	RO
dot1dTpPortMaxInfo	M	RO	M	RO
dot1dTpPortInFrames	M	RO	M	RO
dot1dTpPortOutFrames	M	RO	M	RO
dot1dTpPortInDiscards	M	RO	M	RO
<b>dot1dStaticTable</b>				
Note: Implementation of dot1dStaticTable is OPTIONAL				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
dot1dStaticAddress	O	RW	O	RW
dot1dStaticReceivePort	O	RW	O	RW
dot1dStaticAllowedToGoTo	O	RW	O	RW
dot1dStaticStatus	O	RW	O	RW
<b>DOCS-CABLE-DEVICE-MIB (RFC 2669)</b>				
<b>docsDevBaseGroup</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsDevRole	M	RO	O	RO
docsDevDateTime	M	RW	O	RW
docsDevResetNow	M	RW	O	RW
docsDevSerialNumber	M	RO	O	RO
docsDevSTPControl	M	RW/RO	O	RW/RO
<b>docsDevNmAccessGroup</b>				
NOTE: This group is ONLY required for CM which does not implement SNMPv3 or later.				

<b>docsDevNmAccessTable</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsDevNmAccessIndex	M	N-Acc	O	N-Acc
docsDevNmAccessIp	M	RC	O	RC
docsDevNmAccessIpMask	M	RC	O	RC
docsDevNmAccessCommunity	M	RC	O	RC
docsDevNmAccessControl	M	RC	O	RC
docsDevNmAccessInterfaces	M	RC	O	RC
docsDevNmAccessStatus	M	RC	O	RC
DocsDevNmAccessTrapVersion (Note: This object is currently not in RFC 2669)	M	RC	O	RC
<b>docsDevSoftwareGroup</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsDevSwServer	M	RW	O	RW
docsDevSwFilename	M	RW	O	RW
docsDevSwAdminStatus	M	RW	O	RW
docsDevSwOperStatus	M	RO	O	RO
docsDevSwCurrentVers	M	RO	O	RO
<b>docsDevServerGroup</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsDevServerBootState	M	RO	N-Sup	
docsDevServerDhcp	M	RO	N-Sup	
docsDevServerTime	M	RO	N-Sup	
docsDevServerTftp	M	RO	N-Sup	
docsDevServerConfigFile	M	RO	N-Sup	
<b>docsDevEventGroup</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsDevEvControl	M	RW	M	RW
docsDevEvSyslog	M	RW	M	RW
docsDevEvThrottleAdminStatus	M	RW	M	RW
docsDevEvThrottleInhibited	M	RO	M	RO
docsDevEvThrottleThreshold	M	RW	M	RW
docsDevEvThrottleInterval	M	RW	M	RW
<b>docsDevEvControlTable</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsDevEvPriority	M	N-Acc	M	N-Acc
docsDevEvReporting (Mandatory RW by DOCSIS 1.1; exception to RFC-2669)	M	RW	M	RW

<b>docsDevEventTable</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsDevEvIndex	M	N-Acc	M	N-Acc
docsDevEvFirstTime	M	RO	M	RO
docsDevEvLastTime	M	RO	M	RO
docsDevEvCounts	M	RO	M	RO
docsDevEvLevel	M	RO	M	RO
docsDevEvId	M	RO	M	RO
docsDevEvText	M	RO	M	RO
<b>docsDevFilterGroup</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsDevFilterLLCUnmatchedAction	M	RW	O	RW
<b>docsDevFilterLLCTable</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsDevFilterLLCIndex	M	N-Acc	O	N-Acc
docsDevFilterLLCStatus	M	RC	O	RC
docsDevFilterLLCIfIndex	M	RC	O	RC
docsDevFilterLLCProtocolType	M	RC	O	RC
docsDevFilterLLCProtocol	M	RC	O	RC
docsDevFilterLLCMatches	M	RO	O	RO
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsDevFilterIpDefault	M	RW	O	RW
<b>docsDevFilterIpTable</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsDevFilterIpIndex	M	N-Acc	O	N-Acc
docsDevFilterIpStatus	M	RC	O	RC
docsDevFilterIpControl	M	RC	O	RC
docsDevFilterIpIfIndex	M	RC	O	RC
docsDevFilterIpDirection	M	RC	O	RC
docsDevFilterIpBroadcast	M	RC	O	RC
docsDevFilterIpSaddr	M	RC	O	RC
docsDevFilterIpSmask	M	RC	O	RC
docsDevFilterIpDaddr	M	RC	O	RC
docsDevFilterIpDmask	M	RC	O	RC
docsDevFilterIpProtocol	M	RC	O	RC
docsDevFilterIpSourcePortLow	M	RC	O	RC
docsDevFilterIpSourcePortHigh	M	RC	O	RC
docsDevFilterIpDestPortLow	M	RC	O	RC
docsDevFilterIpDestPortHigh	M	RC	O	RC
docsDevFilterIpMatches	M	RO	O	RO

docsDevFilterIpTos	M	RC	O	RC
docsDevFilterIpTosMask	M	RC	O	RC
docsDevFilterIpContinue	M	RC	O	RC
docsDevFilterIpPolicyId	M	RC	O	RC
<b>docsDevFilterPolicyTable</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsDevFilterPolicyIndex	M	N-Acc	O	N-Acc
docsDevFilterPolicyId	M	RC	O	RC
docsDevFilterPolicyStatus	M	RC	O	RC
docsDevFilterPolicyPtr	M	RC	O	RC
<b>docsDevFilterTosTable</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsDevFilterTosIndex	M	N-Acc	O	N-Acc
docsDevFilterTosStatus	M	RC	O	RC
docsDevFilterTosAndMask	M	RC	O	RC
docsDevFilterTosOrMask	M	RC	O	RC
docsDevCpeGroup				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsDevCpeEnroll	M	RW	N-Sup	
docsDevCpeIpMax	M	RW	N-Sup	
<b>docsDevCpeTable</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsDevCpeIp	M	N-Acc	N-Sup	
docsDevCpeSource	M	RO	N-Sup	
docsDevCpeStatus	M	RC	N-Sup	
<b>IP-MIB (RFC 2011)</b>				
<b>IP Group</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
ipForwarding	M	RW	M	RW
ipDefaultTTL	M	RW	M	RW
ipInreceives	M	RO	M	RO
ipInHdrErrors	M	RO	M	RO
ipInAddrErrors	M	RO	M	RO
ipForwDatagrams	M	RO	M	RO
ipInUnknownProtos	M	RO	M	RO
ipInDiscards	M	RO	M	RO
ipInDelivers	M	RO	M	RO
ipOutRequest	M	RO	M	RO



ipOutDiscard	M	RO	M	RO
ipOutNoRoutes	M	RO	M	RO
ipReasmTimeout	M	RO	M	RO
ipReasmReqds	M	RO	M	RO
ipReasmOKs	M	RO	M	RO
ipReasmFails	M	RO	M	RO
ipFragOKs	M	RO	M	RO
ipFragFails	M	RO	M	RO
ipFragCreates	M	RO	M	RO
<b>ipAddrTable</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
ipAdEntAddr	M	RO	M	RO
ipAdEntIfIndex	M	RO	M	RO
ipAdEntNetMask	M	RO	M	RO
ipAdEntBcastAddr	M	RO	M	RO
ipAdEntReasmMaxSize	M	RO	M	RO
<b>IpNetToMediaTable</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
ipNetToMediaIfIndex	M	RC	M	RC
ipNetToMediaPhysAddress	M	RC	M	RC
ipNetToMediaNetAddress	M	RC	M	RC
ipNetToMediaType	M	RC	M	RC
<b>6.5.1.1.1 Objects</b>				
ipRoutingDiscards	M	RO	M	RO
<b>ICMP Group</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
icmpInMsgs	M	RO	M	RO
icmpInErrors	M	RO	M	RO
icmpInDestUnreachs	M	RO	M	RO
icmpInTimeExcds	M	RO	M	RO
icmpInParmProbs	M	RO	M	RO
icmpInSrcQuenchs	M	RO	M	RO
icmpInRedirects	M	RO	M	RO
icmpInEchos	M	RO	M	RO
icmpInEchosReps	M	RO	M	RO
icmpInTimestamps	M	RO	M	RO
icmpInTimeStampsReps	M	RO	M	RO
icmpInAddrMasks	M	RO	M	RO
icmpInAddrMaskReps	M	RO	M	RO
icmpOutMsgs	M	RO	M	RO

icmpOutErrors	M	RO	M	RO
icmpOutDestUnreachs	M	RO	M	RO
icmpOutTimeExcds	M	RO	M	RO
icmpOutParmProbs	M	RO	M	RO
icmpOutSrcQuenchs	M	RO	M	RO
icmpOutRedirects	M	RO	M	RO
icmpOutEchoes	M	RO	M	RO
icmpOutEchoReps	M	RO	M	RO
icmpOutTimestamps	M	RO	M	RO
icmpOutTimestampReps	M	RO	M	RO
icmpOutAddrMasks	M	RO	M	RO
icmpOutAddrMaskReps	M	RO	M	RO
<b>UDP-MIB (RFC 2013)</b>				
<b>UDP Group</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
udpInDatagrams	M	RO	M	RO
udpNoPorts	M	RO	M	RO
udpInErrors	M	RO	M	RO
udpOutDatagrams	M	RO	M	RO
<b>UDP Listener Table</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
udpLocalAddress	M	RO	M	RO
udpLocalPort	M	RO	M	RO
<b>SNMPv2-MIB (RFC 1907)</b>				
<b>System Group</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
sysDescr	M	RO	M	RO
sysObjectID	M	RO	M	RO
sysUpTime	M	RO	M	RO
sysContact	M	RW	M	RW
sysName	M	RW	M	RW
sysLocation	M	RW	M	RW
sysServices	M	RO	M	RO
sysORLastChange	M	RO	M	RO
<b>sysORTable</b>				
<b>Object</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>

sysORIndex	M	N-Acc	M	N-Acc
sysORID	M	RO	M	RO
sysORDescr	M	RO	M	RO
sysORUpTime	M	RO	M	RO
<b>SNMP Group</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
snmplnPkts	M	RO	M	RO
SnmplnBadVersions	M	RO	M	RO
snmpOutPkts	Ob	RO	Ob	RO
snmplnBadCommunityNames	M	RO	M	RO
snmplnBadCommunityUses	M	RO	M	RO
snmplnASNParseErrs	M	RO	M	RO
snmplnTooBigs	Ob	RO	Ob	RO
snmplnNoSuchNames	Ob	RO	Ob	RO
snmplnBadValues	Ob	RO	Ob	RO
snmplnReadOnlyls	Ob	RO	Ob	RO
snmplnGenErrs	Ob	RO	Ob	RO
snmplnTotalReqVars	Ob	RO	Ob	RO
snmplnTotalSetVars	Ob	RO	Ob	RO
snmplnGetRequests	Ob	RO	Ob	RO
snmplnGetNexts	Ob	RO	Ob	RO
snmplnSetRequests	Ob	RO	Ob	RO
snmplnGetResponses	Ob	RO	Ob	RO
snmplnTraps	Ob	RO	Ob	RO
snmpOutTooBigs	Ob	RO	Ob	RO
snmpOutNoSuchNames	Ob	RO	Ob	RO
snmpOutBadValues	Ob	RO	Ob	RO
snmpOutGenErrs	Ob	RO	Ob	RO
snmpOutGetRequests	Ob	RO	Ob	RO
snmpOutGetNexts	Ob	RO	Ob	RO
snmpOutSetRequests	Ob	RO	Ob	RO
snmpOutGetResponses	Ob	RO	Ob	RO
snmpOutTraps	Ob	RO	Ob	RO
snmpEnableAuthenTraps	M	RW	M	RW
snmpSilentDrops	M	RO	M	RO
snmpProxyDrops	M	RO	M	RO
<b>6.5.1.1.1 Object</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
6.5.1.1.1.2 snmpSetSerialNo	M	RW	M	RW

<b>Etherlike-MIB (RFC 2665)</b>				
<b>dot3StatsTable</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
dot3StatsIndex	M	RO	M	RO
dot3StatsAlignmentErrors	M	RO	M	RO
dot3StatsFCSErrors	M	RO	M	RO
dot3StatsSingleCollisionFrames	M	RO	M	RO
dot3StatsMultipleCollisionFrames	M	RO	M	RO
dot3StatsSQETestErrors	M	RO	M	RO
dot3StatsDeferredTransmissions	M	RO	M	RO
dot3StatsLateCollisions	M	RO	M	RO
dot3StatsExcessiveCollisions	M	RO	M	RO
dot3StatsInternalMacTransmitErrors	M	RO	M	RO
dot3StatsCarrierSenseErrors	M	RO	M	RO
dot3StatsFrameTooLongs	M	RO	M	RO
dot3StatsInternalMacReceiveErrors	M	RO	M	RO
dot3StatsEtherChipSet	D	RO	D	RO
dot3StatsSymbolErrors	M	RO	M	RO
dot3StatsDuplexStatus	M	RO	M	RO
<b>dot3CollTable</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
dot3CollCount	O	NA	O	NA
dot3CollFrequencies	O	RO	O	RO
<b>dot3ControlTable</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
dot3ControlFunctionsSupported	O	RO	O	RO
dot3ControlInUnknownOpcodes	O	RO	O	RO
<b>dot3PauseTable</b>				
<b>Objects</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
dot3PauseAdminMode	O	RW	O	RW
dot3PauseOperMode	O	RO	O	RO
dot3InPauseFrames	O	RO	O	RO
dot3OutPauseFrames	O	RO	O	RO
<b>USB MIB</b>				
<b>NOTE: This MIB is required for CM that supports USB only.</b>				
<b>Object</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
usbNumber	M	RO	NA	

<b>usbPortTable</b>				
<b>Object</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
usbPortIndex	M	RO	NA	
usbPortType	M	RO	NA	
usbPortRate	M	RO	NA	
<b>usbDeviceTable</b>				
<b>Object</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
usbDeviceIndex	M	RO	NA	
usbDevicePower	M	RO	NA	
usbDeviceVendorID	M	RO	NA	
usbDeviceProductID	M	RO	NA	
usbDeviceNumberConfigurations	M	RO	NA	
usbDeviceActiveClass	M	RO	NA	
usbDeviceStatus	M	RO	NA	
usbDeviceEnumCounter	M	RO	NA	
usbDeviceRemoteWakeup	M	RO	NA	
usbDeviceRemoteWakeupOn	M	RO	NA	
<b>usbCDCTable</b>				
<b>Object</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
usbCDCIndex	M	RO	NA	
usbCDCIfIndex	M	RO	NA	
usbCDCSubclass	M	RO	NA	
usbCDCVersion	M	RO	NA	
usbCDCDataTransferType	M	RO	NA	
usbCDCDataEndpoints	M	RO	NA	
usbCDCStalls	M	RO	NA	
<b>usbCDCEtherTable</b>				
<b>Object</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
usbCDCEtherIndex	M	RO	NA	
usbCDCEtherIfIndex	M	RO	NA	
usbCDCEtherMacAddress	M	RO	NA	
usbCDCEtherPacketFilter	M	RO	NA	
usbCDCEtherDataStatisticsCapabilities	M	RO	NA	
usbCDCEtherDataCheckErrs	M	RO	NA	
<b>DOCS-QOS-MIB (draft-ietf-ipcdn-qos-mib-02.txt)</b>				
<b>NOTE: 1.1 CM in 1.0 mode MUST NOT support this MIB.</b>				
<b>docsQosPktClassTable</b>				
<b>Object</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsQosPktClassId	M	N-Acc	M	N-Acc

docsQosPktClassDirection	M	RO	M	RO
docsQosPktClassPriority	M	RO	M	RO
docsQosPktClassIpTosLow	M	RO	M	RO
docsQosPktClassIpTosHigh	M	RO	M	RO
docsQosPktClassIpTosMask	M	RO	M	RO
docsQosPktClassIpProtocol	M	RO	M	RO
docsQosPktClassIpSourceAddr	M	RO	M	RO
docsQosPktClassIpSourceMask	M	RO	M	RO
docsQosPktClassIpDestAddr	M	RO	M	RO
docsQosPktClassIpDestMask	M	RO	M	RO
docsQosPktClassSourcePortStart	M	RO	M	RO
docsQosPktClassSourcePortEnd	M	RO	M	RO
docsQosPktClassDestPortStart	M	RO	M	RO
docsQosPktClassDestPortEnd	M	RO	M	RO
docsQosPktClassDestMacAddr	M	RO	M	RO
docsQosPktClassDestMacMask	M	RO	M	RO
docsQosPktClassSourceMacAddr	M	RO	M	RO
docsQosPktClassEnetProtocolType	M	RO	M	RO
docsQosPktClassEnetProtocol	M	RO	M	RO
docsQosPktClassUserPriApplies	M	RO	M	RO
docsQosPktClassUserPriLow	M	RO	M	RO
docsQosPktClassUserPriHigh	M	RO	M	RO
docsQosPktClassVlanId	M	RO	M	RO
docsQosPktClassState	M	RO	M	RO
docsQosPktClassPkts	M	RO	M	RO

**DocsQosParamSetTable when docsQosParamSetRowType = serviceFlow (1)**

Object	1.1 CM in 1.1 mode	Access	CMTS	Access
docsQosParamSetRowType = <b>serviceFlow (1)</b>	M	N-Acc	M	N-Acc
docsQosParamSetIndex	M	N-Acc	M	N-Acc
docsQosParamSetRowStatus	M	RO	M	RO
docsQosParamSetServiceClassName	M	RO	M	RO
docsQosParamSetPriority	M	RO	M	RO
docsQosParamSetMaxTrafficRate	M	RO	M	RO
docsQosParamSetMaxTrafficBurst	M	RO	M	RO
docsQosParamSetMinReservedRate	M	RO	M	RO
docsQosParamSetMinReservedPkt	M	RO	M	RO
docsQosParamSetActiveTimeout	M	RO	M	RO
docsQosParamSetAdmittedTimeout	M	RO	M	RO
docsQosParamSetMaxConcatBurst	M	RO	M	RO
DocsQosParamSetSchedulingType	M	RO	M	RO
docsQosParamSetRequestPolicy	M	RO	M	RO
docsQosParamSetNomPollInterval	M	RO	M	RO
docsQosParamSetTolPollJitter	M	RO	M	RO
docsQosParamSetUnsolicitGrantSize	M	RO	M	RO
docsQosParamSetNomGrantInterval	M	RO	M	RO

docsQosParamSetTolGrantJitter	M	RO	M	RO
docsQosParamSetGrantsPerInterval	M	RO	M	RO
docsQosParamSetTosAndMask	M	RO	M	RO
docsQosParamSetTosOrMask	M	RO	M	RO
docsQosParamSetMaxLatency	M	RO	M	RO
<b>docsQosParamSetTable</b> when <b>docsQosParamSetRowType = serviceClass (2).</b>				
<b>Object</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
DocsQosParamSetRowType = <b>serviceClass (2)</b>	M	N-Acc	M	N-Acc
docsQosParamSetIndex	M	N-Acc	M	N-Acc
docsQosParamSetRowStatus	M	RO	M	RC
docsQosParamSetServiceClassName	M	RO	M	RO
docsQosParamSetPriority	M	RO	M	RC
docsQosParamSetMaxTrafficRate	M	RO	M	RC
docsQosParamSetMaxTrafficBurst	M	RO	M	RC
docsQosParamSetMinReservedRate	M	RO	M	RC
docsQosParamSetMinReservedPkt	M	RO	M	RC
docsQosParamSetActiveTimeout	M	RO	M	RC
docsQosParamSetAdmittedTimeout	M	RO	M	RC
docsQosParamSetMaxConcatBurst	M	RO	M	RC
DocsQosParamSetSchedulingType	M	RO	M	RC
docsQosParamSetRequestPolicy	M	RO	M	RC
docsQosParamSetNomPollInterval	M	RO	M	RC
docsQosParamSetTolPollJitter	M	RO	M	RC
docsQosParamSetUnsolicitGrantSize	M	RO	M	RC
docsQosParamSetNomGrantInterval	M	RO	M	RC
docsQosParamSetTolGrantJitter	M	RO	M	RC
docsQosParamSetGrantsPerInterval	M	RO	M	RC
docsQosParamSetTosAndMask	M	RO	M	RC
docsQosParamSetTosOrMask	M	RO	M	RC
docsQosParamSetMaxLatency	M	RO	M	RC
<b>docsQosServiceFlowTable</b>				
<b>Object</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsQosServiceFlowId	M	N-Acc	M	N-Acc
docsQosServiceFlowProvisionedParamSetIndex	M	RO	M	RO
docsQosServiceFlowAdmittedParamSetIndex	M	RO	M	RO
docsQosServiceFlowActiveParamSetIndex	M	RO	M	RO
docsQosServiceFlowSID	M	RO	M	RO
docsQosServiceFlowDirection	M	RO	M	RO
docsQosServiceFlowPrimary	M	RO	M	RO
<b>DocsQosServiceFlowStatsTable</b>				
<b>Object</b>	<b>1.1 CM in</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>

	<b>1.1 mode</b>			
docsQosServiceFlowPkts	M	RO	M	RO
docsQosServiceFlowOctets	M	RO	M	RO
docsQosServiceFlowTimeCreated	M	RO	M	RO
docsQosServiceFlowTimeActive	M	RO	M	RO
docsQosServiceFlowPHSUnknowns	M	RO	M	RO
docsQosServiceFlowPolicedDropPkts	M	RO	M	RO
docsQosServiceFlowPolicedDelayPkts	M	RO	M	RO
<b>docsQosUpstreamStatsTable</b>				
<b>Object</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsQosSID	N-Sup		M	N-Acc
docsQosUpstreamFragPkts	N-Sup		M	RO
docsQosUpstreamIncompletePkts	N-Sup		M	RO
docsQosUpstreamConcatBursts	N-Sup		M	RO
<b>docsQosDynamicServiceStatsTable</b>				
<b>Object</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsQosIfDirection	M	N-Acc	M	N-Acc
docsQosDSAReqs	M	RO	M	RO
docsQosDSARsps	M	RO	M	RO
docsQosDSAAcks	M	RO	M	RO
DocsQosDSCReqs	M	RO	M	RO
docsQosDSCRsps	M	RO	M	RO
docsQosDSCAcks	M	RO	M	RO
docsQosDSDReqs	M	RO	M	RO
docsQosDSDRsps	M	RO	M	RO
docsQosDynamicAdds	M	RO	M	RO
docsQosDynamicAddFails	M	RO	M	RO
DocsQosDynamicChanges	M	RO	M	RO
docsQosDynamicChangeFails	M	RO	M	RO
DocsQosDynamicDeletes	M	RO	M	RO
DocsQosDynamicDeleteFails	M	RO	M	RO
<b>docsQosServiceFlowLogTable</b>				
<b>Object</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsQosServiceFlowLogIndex	N-Sup		M	N-Acc
docsQosServiceFlowLogIfIndex	N-Sup		M	RO
docsQosServiceFlowLogSFID	N-Sup		M	RO
docsQosServiceFlowLogCmMac	N-Sup		M	RO
docsQosServiceFlowLogPkts	N-Sup		M	RO
docsQosServiceFlowLogOctets	N-Sup		M	RO
docsQosServiceFlowLogTimeDeleted	N-Sup		M	RO
DocsQosServiceFlowLogTimeCreated	N-Sup		M	RO
docsQosServiceFlowLogTimeActive	N-Sup		M	RO



DocsQosServiceFlowLogControl	N-Sup		M	RW
<b>docsQosServiceClassTable</b>				
<b>Object</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsQosServiceClassName	N-Sup		M	N-Acc
docsQosServiceClassParamSetIndex	N-Sup		M	RC
docsQosServiceClassStatus	N-Sup		M	RC
<b>docsQosServiceClassPolicyTable</b>				
<b>Object</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsQosServiceClassPolicyIndex	O	N-Acc	O	N-Acc
docsQosServiceClassPolicyName	O	RC	O	RC
docsQosServiceClassPolicyRulePriority	O	RC	O	RC
docsQosServiceClassPolicyStatus	O	RC	O	RC
<b>docsQosPHSTable</b>				
<b>Object</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsQosPHSIndex	O	N-Acc	O	N-Acc
docsQosPHSField	O	RO	O	RO
docsQosPHSMask	O	RO	O	RO
docsQosPHSSize	O	RO	O	RO
docsQosPHSVerify	O	RO	O	RO
docsQosPHSClassifierIndex	O	RO	O	RO
<b>docsQosCmtsMacToSrvFlowTable</b>				
<b>Object</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsQosCmtsCmMac	N-Sup		M	N-Acc
docsQosCmtsServiceFlowId	N-Sup		M	N-Acc
docsQosCmtsIfIndex	N-Sup		M	RO
<b>DOCS-SUBMGT-MIB (draft-ietf-ipcdn-subscriber-mib-01.txt) Subscribe Management MIB</b>				
<b>docsSubMgtCpeControlTable</b>				
<b>Object</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsSubMgtCpeControlMaxCpelp	NA	NA	M	RW
docsSubMgtCpeControlActive	NA	NA	M	RW
docsSubMgtCpeControlLearnable	NA	NA	M	RW
docsSubMgtCpeControlReset	NA	NA	M	RW
docsSubMgtCpeMaxIpDefault	NA	NA	M	RW
DocsSubMgtCpeActiveDefault	NA	NA	M	RW
<b>docsSubMgtCpelpTable</b>				

Object	CM	Access	CMTS	Access
DocsSubMgtCpelpIndex	NA	NA	M	N-Acc
DocsSubMgtCpelpAddr	NA	NA	M	RO
DocsSubMgtCpelpLearned	NA	NA	M	RO
<b>docsSubMgtPktFilterTable</b>				
Object	CM	Access	CMTS	Access
DocsSubMgtPktFilterGroup	NA	NA	M	N-Acc
docsSubMgtPktFilterIndex	NA	NA	M	N-Acc
docsSubMgtPktFilterSrcAddr	NA	NA	M	RC
docsSubMgtPktFilterSrcMask	NA	NA	M	RC
DocsSubMgtPktFilterDstAddr	NA	NA	M	RC
docsSubMgtPktFilterDstMask	NA	NA	M	RC
DocsSubMgtPktFilterUlp	NA	NA	M	RC
docsSubMgtPktFilterTosValue	NA	NA	M	RC
docsSubMgtPktFilterTosMask	NA	NA	M	RC
DocsSubMgtPktFilterAction	NA	NA	M	RC
docsSubMgtPktFilterMatches	NA	NA	M	RO
docsSubMgtPktFilterStatus	NA	NA	M	RC
<b>docsSubMgtTcpUdpFilterTable</b>				
Object	CM	Access	CMTS	Access
DocsSubMgtTcpUdpSrcPort	NA	NA	M	RC
docsSubMgtTcpUdpDstPort	NA	NA	M	RC
docsSubMgtTcpFlagValues	NA	NA	M	RC
docsSubMgtTcpFlagMask	NA	NA	M	RC
DocsSubMgtTcpUdpStatus	NA	NA	M	RC
<b>docsSubMgtCmFilterTable</b>				
Object	CM	Access	CMTS	Access
docsSubMgtSubFilterDownstream	NA	NA	M	RW
docsSubMgtSubFilterUpstream	NA	NA	M	NW
docsSubMgtCmFilterDownstream	NA	NA	M	RW
DocsSubMgtCmFilterUpstream	NA	NA	M	RW
Object	CM	Access	CMTS	Access
docsSubMgtSubFilterDownDefault	NA	NA	M	RW
DocsSubMgtSubFilterUpDefault	NA	NA	M	RW
docsSubMgtCmFilterDownDefault	NA	NA	M	RW
docsSubMgtCmFilterUpDefault	NA	NA	M	RW
<b>IGMP-STD-MIB (draft-ietf-idmr-igmp-mib-13.txt draft-ietf-ipcdn-igmp-mib-01.txt)</b>				
This MIB is optional for Bridging CMTS				
<b>NOTE: 1.1 CM in 1.0 mode MUST NOT support this MIB.</b>				
<b>IgmplInterfaceTable</b>				

Object	1.1 CM in 1.1 mode	Access	CMTS	Access
igmpInterfaceIfIndex	M	N-Acc	M	N-Acc
igmpInterfaceQueryInterval	M	RC	M	RC
igmpInterfaceStatus	M	RC	M	RC
igmpInterfaceVersion	M	RC	M	RC
igmpInterfaceQuerier	M	RO	M	RO
igmpInterfaceQueryMaxResponseTime	M	RO	M	RO
igmpInterfaceVersion1QuerierTimer	M	RO	M	RO
igmpInterfaceWrongVersionQueries	M	RO	M	RO
igmpInterfaceJoins	M	RO	M	RO
igmpInterfaceGroups	M	RO	M	RO
igmpInterfaceRobustness	M	RC	M	RC
igmpInterfaceLastMembQueryIntvl	M	RC	M	RC
igmpInterfaceProxyIfIndex	M	RC	M	RC
igmpInterfaceQuerierUpTime	M	RO	M	RO
igmpInterfaceQuerierExpiryTime	M	RO	M	RO
<b>igmpCacheTable</b>				
Object	1.1 CM in 1.1 mode	Access	CMTS	Access
igmpCacheAddress	M	N-Acc	M	N-Acc
igmpCacheIfIndex	M	N-Acc	M	N-Acc
igmpCacheSelf	M	RC	M	RC
igmpCacheLastReporter	M	RO	M	RO
igmpCacheUpTime	M	RO	M	RO
igmpCacheExpiryTime	M	RO	M	RO
igmpCacheStatus	M	RC	M	RC
igmpCacheVersion1HostTimer	M	RO	M	RO
<b>Account Management MIB (MIB defining work is still in progress.)</b>				
<b>docsCpeSegmentTable</b>				
Object	CM	Access	CMTS	Access
docsCpeSegmentID	NA	NA	O	RO
docsCpeSegmentIp	NA	NA	O	RC
<b>DocsCpeTrafficData Table</b>				
Object	CM	Access	CMTS	Access
docsCpeIpAddress	NA	NA	O	RO
docsCpeTrafficDataUpStreamPackets	NA	NA	O	RC
docsCpeTrafficDataDownStreamPackets	NA	NA	O	RC
docsCpeTrafficDataUpStreamOctets	NA	NA	O	RC
docsCpeTrafficDataDownStreamOctets	NA	NA	O	RC
docsCpeTrafficDataUpStreamDropPackets	NA	NA	O	RC

docsCpeTrafficDataDownStreamDropPackets	NA	NA	O	RC
<b>docsCmCpeTable</b>	<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsCmMacAddress	NA	NA	O	RC
docsCmIpAddress	NA	NA	O	RC
docsCpeMACAddress	NA	NA	O	RC
docsCpelpAddress	NA	NA	O	RC

Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
<b>DOCS-BPI-MIB (draft-ietf-ipcdn-mcns-bpi-mib-01.txt)</b>						
<b>docsBpiCmBaseTable</b>						
Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docsBpiCmPrivacyEnable	M	RO	N-Sup		NA	
docsBpiCmPublicKey	M	RO	N-Sup		NA	
docsBpiCmAuthState	M	RO	N-Sup		NA	
docsBpiCmAuthKeySequenceNumber	M	RO	N-Sup		NA	
docsBpiCmAuthExpires	M	RO	N-Sup		NA	
docsBpiCmAuthReset	M	RW	N-Sup		NA	
docsBpiCmAuthGraceTime	M	RO	N-Sup		NA	
docsBpiCmTEKGraceTime	M	RO	N-Sup		NA	
docsBpiCmAuthWaitTimeout	M	RO	N-Sup		NA	
docsBpiCmReauthWaitTimeout	M	RO	N-Sup		NA	
docsBpiCmOpWaitTimeout	M	RO	N-Sup		NA	
docsBpiCmRekeyWaitTimeout	M	RO	N-Sup		NA	
docsBpiCmAuthRejectWaitTimeout	M	RO	N-Sup		NA	
docsBpiCmAuthRequests	M	RO	N-Sup		NA	
docsBpiCmAuthReplies	M	RO	N-Sup		NA	
docsBpiCmAuthRejects	M	RO	N-Sup		NA	
docsBpiCmAuthInvalids	M	RO	N-Sup		NA	
docsBpiCmAuthRejectErrorCode	M	RO	N-Sup		NA	
docsBpiCmAuthRejectErrorString	M	RO	N-Sup		NA	

docsBpiCmAuthInvalidErrorCode	M	RO	N-Sup		NA	
docsBpiCmAuthInvalidErrorString	M	RO	N-Sup		NA	
<b>docsBpiCmTEKTable</b>						
<b>Object</b>	<b>1.1CM in 1.0 mode</b>	<b>Access</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsBpiCmTEKPrivacyEnable	M	RO	N-Sup		NA	
docsBpiCmTEKState	M	RO	N-Sup		NA	
docsBpiCmTEKExpiresOld	M	RO	N-Sup		NA	
docsBpiCmTEKExpiresNew	M	RO	N-Sup		NA	
docsBpiCmTEKKeyRequests	M	RO	N-Sup		NA	
docsBpiCmTEKKeyReplies	M	RO	N-Sup		NA	
docsBpiCmTEKKeyRejects	M	RO	N-Sup		NA	
docsBpiCmTEKInvalids	M	RO	N-Sup		NA	
docsBpiCmTEKAuthPends	M	RO	N-Sup		NA	
docsBpiCmTEKKeyRejectErrorCode	M	RO	N-Sup		NA	
docsBpiCmTEKKeyRejectErrorString	M	RO	N-Sup		NA	
docsBpiCmTEKInvalidErrorCode	M	RO	N-Sup		NA	
docsBpiCmTEKInvalidErrorString	M	RO	N-Sup		NA	
<b>docsBpiCmtsBaseTable</b>						
<b>Object</b>	<b>1.1CM in 1.0 mode</b>	<b>Access</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsBpiCmtsDefaultAuthLifetime	NA		NA		N-Sup	
docsBpiCmtsDefaultTEKLifetime	NA		NA		N-Sup	
docsBpiCmtsAuthRequests	NA		NA		N-Sup	
docsBpiCmtsAuthReplies	NA		NA		N-Sup	
docsBpiCmtsAuthRejects	NA		NA		N-Sup	
docsBpiCmtsAuthInvalids	NA		NA		N-Sup	
<b>docsBpiCmtsAuthTable</b>						
<b>Object</b>	<b>1.1CM in 1.0 mode</b>	<b>Access</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsBpiCmtsAuthCmMacAddress	NA		NA		N-Sup	
docsBpiCmtsAuthCmPublicKey	NA		NA		N-Sup	
docsBpiCmtsAuthCmKeySequenceNumber	NA		NA		N-Sup	
docsBpiCmtsAuthCmExpires	NA		NA		N-Sup	
docsBpiCmtsAuthCmLifetime	NA		NA		N-Sup	
docsBpiCmtsAuthCmGraceTime	NA		NA		N-Sup	
docsBpiCmtsAuthCmReset	NA		NA		N-Sup	
docsBpiCmtsAuthCmRequests	NA		NA		N-Sup	
docsBpiCmtsAuthCmReplies	NA		NA		N-Sup	
docsBpiCmtsAuthCmRejects	NA		NA		N-Sup	
docsBpiCmtsAuthCmInvalids	NA		NA		N-Sup	
docsBpiCmtsAuthRejectErrorCode	NA		NA		N-Sup	

docsBpiCmtsAuthRejectErrorString	NA		NA		N-Sup	
docsBpiCmtsAuthInvalidErrorCode	NA		NA		N-Sup	
docsBpiCmtsAuthInvalidErrorString	NA		NA		N-Sup	
<b>docsBpiCmtsTEKTable</b>						
<b>Object</b>	<b>1.1CM in 1.0 mode</b>	<b>Access</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsBpiCmtsTEKLifetime	NA		NA		N-Sup	
docsBpiCmtsTEKGraceTime	NA		NA		N-Sup	
docsBpiCmtsTEKExpiresOld	NA		NA		N-Sup	
docsBpiCmtsTEKExpiresNew	NA		NA		N-Sup	
docsBpiCmtsTEKReset	NA		NA		N-Sup	
docsBpiCmtsKeyRequests	NA		NA		N-Sup	
docsBpiCmtsKeyReplies	NA		NA		N-Sup	
docsBpiCmtsKeyRejects	NA		NA		N-Sup	
docsBpiCmtsTEKInvalids	NA		NA		N-Sup	
docsBpiCmtsKeyRejectErrorCode	NA		NA		N-Sup	
docsBpiCmtsKeyRejectErrorString	NA		NA		N-Sup	
docsBpiCmtsTEKInvalidErrorCode	NA		NA		N-Sup	
docsBpiCmtsTEKInvalidErrorString	NA		NA		N-Sup	
<b>docsBpiIpMulticastMapTable</b>						
<b>Object</b>	<b>1.1CM in 1.0 mode</b>	<b>Access</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsBpiIpMulticastAddress	NA		NA		N-Sup	
docsBpiIpMulticastprefixLength	NA		NA		N-Sup	
docsBpiIpMulticastServiceId	NA		NA		N-Sup	
docsBpiIpMulticastMapControl	NA		NA		N-Sup	
<b>docsBpiMulticastAuthTable</b>						
<b>Object</b>	<b>1.1CM in 1.0 mode</b>	<b>Access</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsBpiMulticastServiceId	NA		NA		N-Sup	
docsBpiMulticastCmMacAddress	NA		NA		N-Sup	
docsBpiMulticastAuthControl	NA		NA		N-Sup	
<b>DOCS-BPI2-MIB (draft-ietf-ipcdn-bpiplus-mib-02.txt)</b>						
<b>docsBpi2CmBaseTable</b>						
<b>Object</b>	<b>1.1CM in 1.0 mode</b>	<b>Access</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsBpi2CmPrivacyEnable	O	RO	M	RO	NA	
docsBpi2CmPublicKey	O	RO	M	RO	NA	
docsBpi2CmAuthState	O	RO	M	RO	NA	
docsBpi2CmAuthKeySequenceNumber	O	RO	M	RO	NA	

docsBpi2CmAuthExpiresOld	O	RO	M	RO	NA	
docsBpi2CmAuthExpiresNew	O	RO	M	RO	NA	
docsBpi2CmAuthReset	O	RW	M	RW	NA	
docsBpi2CmAuthGraceTime	O	RO	M	RO	NA	
docsBpi2CmTEKGraceTime	O	RO	M	RO	NA	
docsBpi2CmAuthWaitTimeout	O	RO	M	RO	NA	
docsBpi2CmReauthWaitTimeout	O	RO	M	RO	NA	
docsBpi2CmOpWaitTimeout	O	RO	M	RO	NA	
docsBpi2CmRekeyWaitTimeout	O	RO	M	RO	NA	
docsBpi2CmAuthRejectWaitTimeout	O	RO	M	RO	NA	
docsBpi2CmSAMapWaitTimeout	O	RO	M	RO	NA	
docsBpi2CmSAMapMaxRetries	O	RO	M	RO	NA	
docsBpi2CmAuthentInfos	O	RO	M	RO	NA	
docsBpi2CmAuthRequests	O	RO	M	RO	NA	
docsBpi2CmAuthReplies	O	RO	M	RO	NA	
docsBpi2CmAuthRejects	O	RO	M	RO	NA	
docsBpi2CmAuthInvalids	O	RO	M	RO	NA	
docsBpi2CmAuthRejectErrorCode	O	RO	M	RO	NA	
docsBpi2CmAuthRejectErrorString	O	RO	M	RO	NA	
<b>docsBpi2CmTEKTable</b>						
<b>Object</b>	<b>1.1CM in 1.0 mode</b>	<b>Access</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsBpi2CmTEKSAId	O	RO	M	RO	NA	
docsBpi2CmTEKSAType	O	RO	M	RO	NA	
docsBpi2CmTEKDataEncryptAlg	O	RO	M	RO	NA	
docsBpi2CmTEKDataAuthentAlg	O	RO	M	RO	NA	
docsBpi2CmTEKState	O	RO	M	RO	NA	
docsBpi2CmTEKKeySequenceNumber	O	RO	M	RO	NA	
docsBpi2CmTEKExpiresOld	O	RO	M	RO	NA	
docsBpi2CmTEKExpiresNew	O	RO	M	RO	NA	
docsBpi2CmTEKKeyRequests	O	RO	M	RO	NA	
docsBpi2CmTEKKeyReplies	O	RO	M	RO	NA	
docsBpi2CmTEKKeyRejects	O	RO	M	RO	NA	
docsBpi2CmTEKInvalids	O	RO	M	RO	NA	
docsBpi2CmTEKAuthPends	O	RO	M	RO	NA	
docsBpi2CmTEKKeyRejectErrorCode	O	RO	M	RO	NA	
docsBpi2CmTEKKeyRejectErrorString	O	RO	M	RO	NA	
docsBpi2CmTEKInvalidErrorCode	O	RO	M	RO	NA	
docsBpi2CmTEKInvalidErrorString	O	RO	M	RO	NA	
<b>docsBpi2CmIplMulticastMapTable</b>						
<b>Object</b>	<b>1.1CM in 1.0 mode</b>	<b>Access</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsBpi2CmIplMulticastAddress	O	N-Acc	M	N-Acc	NA	
docsBpi2CmIplMulticastSAId	O	RO	M	RO	NA	

docsBpi2CmIpMulticastSAMapState	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapRequests	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapReplies	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapRejects	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapRejectErrorCode	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapRejectErrorString	O	RO	M	RO	NA	
<b>docsBpi2CmDeviceCertTable</b>						
<b>Object</b>	<b>1.1CM in 1.0 mode</b>	<b>Access</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsBpi2CmDeviceCmCert	M	RW	M	RW	NA	
docsBpi2CmDeviceManufCert	M	RO	M	RO	NA	
<b>docsBpi2CmCryptoSuiteTable</b>						
<b>Object</b>	<b>1.1CM in 1.0 mode</b>	<b>Access</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsBpi2CmCryptoSuiteIndex	M	N-Acc	M	N-Acc	NA	
docsBpi2CmCryptoSuiteDataEncryptAlg	M	RO	M	RO	NA	
docsBpi2CmCryptoSuiteDataAuthentAlg	M	RO	M	RO	NA	
<b>docsBpi2CmtsBaseEntryTable</b>						
<b>Object</b>	<b>1.1CM in 1.0 mode</b>	<b>Access</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsBpi2CmtsDefaultAuthLifetime	NA		NA		M	RW
docsBpi2CmtsDefaultTEKLifetime	NA		NA		M	RW
docsBpi2CmtsDefaultSelfSignedManufCertTrust	NA		NA		M	RW
docsBpi2CmtsCheckCertValidityPeriods	NA		NA		M	RW
docsBpi2CmtsAuthentInfos	NA		NA		M	RO
docsBpi2CmtsAuthRequests	NA		NA		M	RO
docsBpi2CmtsAuthReplies	NA		NA		M	RO
docsBpi2CmtsAuthRejects	NA		NA		M	RO
docsBpi2CmtsAuthInvalids	NA		NA		M	RO
docsBpi2CmtsSAMapRequests	NA		NA		M	RO
docsBpi2CmtsSAMapReplies	NA		NA		M	RO
docsBpi2CmtsSAMapRejects	NA		NA		M	RO
<b>docsBpi2CmtsAuthEntryTable</b>						
<b>Object</b>	<b>1.1CM in 1.0 mode</b>	<b>Access</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsBpi2CmtsAuthCmMacAddress	NA		NA		M	N-Acc
docsBpi2CmtsAuthCmBpiVersion	NA		NA		M	RO



docsBpi2CmtsAuthCmPublicKey	NA		NA		M	RO
docsBpi2CmtsAuthCmKeySequenceNumber	NA		NA		M	RO
docsBpi2CmtsAuthCmExpiresOld	NA		NA		M	RO
docsBpi2CmtsAuthCmExpiresNew	NA		NA		M	RO
docsBpi2CmtsAuthCmLifetime	NA		NA		M	RW
docsBpi2CmtsAuthCmGraceTime	NA		NA		M	RO
docsBpi2CmtsAuthCmReset	NA		NA		M	RW
docsBpi2CmtsAuthCmInfos	NA		NA		M	RO
docsBpi2CmtsAuthCmRequests	NA		NA		M	RO
docsBpi2CmtsAuthCmReplies	NA		NA		M	RO
docsBpi2CmtsAuthCmRejects	NA		NA		M	RO
docsBpi2CmtsAuthCmInvalids	NA		NA		M	RO
docsBpi2CmtsAuthRejectErrorCode	NA		NA		M	RO
docsBpi2CmtsAuthRejectErrorString	NA		NA		M	RO
docsBpi2CmtsAuthInvalidErrorCode	NA		NA		M	RO
docsBpi2CmtsAuthInvalidErrorString	NA		NA		M	RO
docsBpi2CmtsAuthPrimarySAId	NA		NA		M	RO
docsBpi2CmtsAuthBpkmCmCertValid	NA		NA		M	RO
docsBpi2CmtsAuthBpkmCmCert	NA		NA		M	RO
<b>docsBpi2CmtsTEKTable</b>						
<b>Object</b>	<b>1.1CM in 1.0 mode</b>	<b>Access</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsBpi2CmtsTEKSAId	NA		NA		M	N-Acc
docsBpi2CmtsTEKSAType	NA		NA		M	RO
docsBpi2CmtsTEKDataEncryptAlg	NA		NA		M	RO
docsBpi2CmtsTEKDataAuthentAlg	NA		NA		M	RO
docsBpi2CmtsTEKLifetime	NA		NA		M	RW
docsBpi2CmtsTEKGraceTime	NA		NA		M	RO
docsBpi2CmtsTEKKeySequenceNumber	NA		NA		M	RO
docsBpi2CmtsTEKExpiresOld	NA		NA		M	RO
docsBpi2CmtsTEKExpiresNew	NA		NA		M	RO
docsBpi2CmtsTEKReset	NA		NA		M	RW
docsBpi2CmtsKeyRequests	NA		NA		M	RO
docsBpi2CmtsKeyReplies	NA		NA		M	RO
docsBpi2CmtsKeyRejects	NA		NA		M	RO
docsBpi2CmtsTEKInvalids	NA		NA		M	RO
docsBpi2CmtsKeyRejectErrorCode	NA		NA		M	RO
docsBpi2CmtsKeyRejectErrorString	NA		NA		M	RO
docsBpi2CmtsTEKInvalidErrorCode	NA		NA		M	RO
docsBpi2CmtsTEKInvalidErrorString	NA		NA		M	RO

<b>docsBpi2CmtsIpMulticastMapTable</b>						
<b>Object</b>	<b>1.1CM in 1.0 mode</b>	<b>Access</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsBpi2CmtsIpMulticastAddress	NA		NA		M	N-Acc
docsBpi2CmtsIpMulticastPrefixLength	NA		NA		M	N-Acc
docsBpi2CmtsIpMulticastSAId	NA		NA		M	RC/RO
docsBpi2CmtsIpMulticastSAMapRequests	NA		NA		M	RO
docsBpi2CmtsIpMulticastSAMapReplies	NA		NA		M	RO
docsBpi2CmtsIpMulticastSAMapRejects	NA		NA		M	RO
docsBpi2CmtsIpMulticastSAMapRejectErrorCode	NA		NA		M	RO
docsBpi2CmtsIpMulticastSAMapRejectErrorString	NA		NA		M	RO
docsBpi2CmtsIpMulticastMapControl	NA		NA		M	RC/RO
<b>docsBpi2CmtsMulticastAuthTable</b>						
<b>Object</b>	<b>1.1CM in 1.0 mode</b>	<b>Access</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsBpi2CmtsMulticastAuthSAId	NA		NA		M	N-Acc
docsBpi2CmtsMulticastAuthCmMacAddress	NA		NA		M	N-Acc
docsBpi2CmtsMulticastAuthControl	NA		NA		M	RC/RO
<b>docsBpi2CmtsProvisionedCmCertificateTable</b>						
<b>Object</b>	<b>1.1CM in 1.0 mode</b>	<b>Access</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsBpi2CmtsProvisionedCmCertificateMacAddress	NA		NA		M	N-Acc
docsBpi2CmtsProvisionedCmCertificateTrust	NA		NA		M	RC
docsBpi2CmtsProvisionedCmCertificateSource	NA		NA		M	RO
docsBpi2CmtsProvisionedCmCertificateStatus	NA		NA		M	RC
docsBpi2CmtsProvisionedCmCertificate}	NA		NA		M	RC
<b>docsBpi2CmtsCACertificateTable</b>						
<b>Object</b>	<b>1.1CM in 1.0 mode</b>	<b>Access</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsBpi2CmtsCACertificateIndex	NA		NA		M	N-Acc
docsBpi2CmtsCACertificateSubject	NA		NA		M	RO
docsBpi2CmtsCACertificateIssuer	NA		NA		M	RO
docsBpi2CmtsCACertificateSerialNumber	NA		NA		M	RO
docsBpi2CmtsCACertificateTrust	NA		NA		M	RC
docsBpi2CmtsCACertificateSource	NA		NA		M	RO
docsBpi2CmtsCACertificateStatus	NA		NA		M	RC

docsBpi2CmtsCACert	NA		NA		M	RC
<b>docsBpi2CodeDownloadGroup</b>						
<b>Object</b>	<b>1.1CM in 1.0 mode</b>	<b>Access</b>	<b>1.1 CM in 1.1 mode</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
docsBpi2CodeDownloadStatusCode,	M	RO	M	RO	O	RO
docsBpi2CodeMfgCodeDownloadStatusString,	M	RO	M	RO	O	RO
docsBpi2CodeMfgOrgName,	M	RO	M	RO	O	RO
docsBpi2CodeMfgCodeAccessStart,	M	RO	M	RO	O	RO
docsBpi2CodeMfgCvcAccessStart,	M	RO	M	RO	O	RO
docsBpi2CodeCoSignerOrgName,	M	RO	M	RO	O	RO
docsBpi2CodeCoSignerCodeAccessStart,	M	RO	M	RO	O	RO
docsBpi2CodeCoSignerCvcAccessStart,	M	RO	M	RO	O	RO
docsBpi2CodeCvcUpdate	M	RW	M	RW	O	RW
<b>SNMP-USM-DH-OBJECTS-MIB (RFC 2786)</b>						
<b>Object</b>			<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
usmDHParameters			M	RW	O	RW
<b>usmDHUserKeyTable</b>						
<b>Object</b>			<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
usmDHUserAuthKeyChange			M	RC	O	RC
UsmDHUserOwnAuthKeyChange			M	RC	O	RC
usmDHUserPrivKeyChange			M	RC	O	RC
usmDHUserOwnPrivKeyChange			M	RC	O	RC
<b>usmDHKickstartTable</b>						
<b>Object</b>			<b>CM</b>	<b>Access</b>	<b>CMTS</b>	<b>Access</b>
usmDHKickstartIndex			M	N-Acc	O	N-Acc
usmDHKickstartMyPublic			M	RO	O	RO
usmDHKickstartMgrPublic			M	RO	O	RO
usmDHKickstartSecurityName			M	RO	O	RO

## **APPENDIX B. Business Process Scenarios For Subscriber Account Management**

In order to develop the DOCS-OSS Subscriber Account Management Specification, it is necessary to consider high-level business processes common to cable operators and the associated operational scenarios. The following definitions represent a generic view of key processes involved. It is understood that business process terminology vary among different cable operators, distinguished by unique operating environments and target market segments

For the purpose of this document, Subscriber Account Management refers to the following business processes and terms:

**Class of Service Provisioning Processes, which are involved in the automatic and dynamic provisioning and enforcement of subscribed class of policy-based service level agreements (SLAs);**

**Usage-Based Billing Processes, which are involved in the processing of bills based on services rendered to and consumed by paying subscriber customers.**

### **B.1. The Old Service Model -- “One Class Only” & “Best Effort” Service**

The Internet is an egalitarian cyber society in its pure technical form where all Internet Protocol (IP) packets are treated as equals. Given all IP packets have equal right of way over the Internet, it is a “one class fits all”, “first come, first serve” type of service level arrangement. The response time and quality of delivery service is promised to be on a “best effort” basis only.

Unfortunately, while all IP packets are theoretically equal, certain classes of IP packets must be processed differently. When transmitting data packets, traffic congestion causes no fatal problems except unpredictable delays and frustrations. However, in a convergent IP world where data packets are mixed with those associated with voice and streaming video, such “one class” service level and “best effort only” quality is not workable.

### **B.2. The Old Billing Model -- “Flat Rate” Access**

As high speed data over cable service deployment moves to the next stage, serious considerations must be made by all cable operators to abandon old business practices, most notably “flat rate” fee structure. No service provider can hope to stay in business long by continuing to offer a single, “flat rate” access service to all subscribers, regardless of actual usage.

Imagine your utility bills were the same month after month, whether you used very little water or electricity every day, or if you ran your water and your air conditioning at full blast 24 hours a day. You are entitled, just like everyone else, to consume as much or as little as you wished, anytime you wanted it. Chances are you would not accept such a service agreement. Not only because it is not a fair arrangement, but also because such wasteful consumption would put pressure on the finite supply of water and electricity that most of your normal demands for usage would likely go unfulfilled.

### **B.3. A Successful New Business Paradigm**

The new paradigm for delivering IP-based services over cable networks is forcing all cable operators to adopt a new business paradigm. The retention of customers will require that an operator offer different class of service options and associated access rates with guaranteed provisioning and delivery of subscribed services. “Back Office” usage-based accounting and subscriber billing will become an important competitive differentiation in the emergence of high-speed data over cable services.

### B.3.1 Integrating “Front End” Processes Seamlessly with “Back Office” Functions

A long-standing business axiom states that accountability exists only with the right measurements and that business prospers only with the proper management information. An effective subscriber account management system for data over cable services should meet three (3) major requirements:

Automatic & Dynamic Subscriber Provisioning

**The 1st requirement is to integrate service subscription orders and changes automatically and dynamically, with the various processes that invoke the provisioning and delivering of subscribed and/or “on demand” services;**

Guaranteed Class & Quality of Services

**The 2nd requirement is to offer different class of services with varying rates and guarantee the quality of service level associated with each service class;**

Data Collection, Warehousing & Usage Billing

**The 3rd requirement is to capture a subscriber’s actual usage, calculating the bill based on the rate associated with the customer’s subscribed service levels.**

### B.3.2 Designing Class of Services

While designing different class of service offerings, a cable operator might consider the following framework:

Class of Service by Account Type – Business vs. Residential Accounts

Class of Service by Guaranteed Service Levels

Class of Service by Time of Day and/or Day of Week

“On Demand” Service by Special Order

The following is a plausible sample of class of services:

- “Best Effort” Service Without Minimum Guarantee  
**This class of “Best Effort Only” service is the normal practice of today where subscribers of this class of service are allocated only excess channel bandwidth available at the time while each subscriber’s access is capped at a maximum bandwidth (for example at 512 kilobit per second).**
- Platinum Service for Business and High-Access Residential Accounts  
**Business accounts subscribing to this service are guaranteed a minimum data rate of downstream bandwidth – 512 kilobit per second – and if excess bandwidth is available, they are allowed to burst to 10 megabit per second.**
- Gold Service for Business Accounts  
**This class of service guarantees subscribers a 256 kilobit per second downstream data rate during business hours (for example from 8 a.m. to 6 p.m.) and 128 kilobit per second at other times. If excess bandwidth is available at any time, data is allowed to burst to 5 megabit per second.**
- Gold Service for Residential Accounts

Residential subscribers of this service are guaranteed 128 kilobit per second downstream bandwidth during business hours and 256 kilobit per second at other times (for example from 6 p.m. to 8 a.m.), and a maximum data burst rate of 5 megabit per second with available excess bandwidth.

- *Silver Service for Business Accounts*

Business accounts subscribing to this service are guaranteed 128 kilobit per second downstream data rate during business hours and 64 kilobit per second during other times, and a maximum burst rate of 1 megabit per second.

- *Silver Service for Residential Accounts*

Subscribers are guaranteed 64 kilobit per second downstream bandwidth during business hours and 128 kilobit per second at other times, with a maximum burst rate of 1 megabit per second.

- *“On Demand” Service by Special Order*

This class of “on demand” service allows a subscriber to request additional bandwidth available for a specific period of time. For example, a subscriber can go to operator’s web site and requests for increased guaranteed bandwidth service levels from his registered subscribed class of service from the normal 256 kilobit per second to 1 megabit per second from 2 p.m. to 4 p.m. the following day only, after which his service levels returns to the original subscribed class. The provisioning server will check the bandwidth commitment and utilization history to decide whether such “on demand” service is granted.

### **B.3.3 Usage-Based Billing**

A complete billing solution involves the following processes:

- Design different usage-based billing options
- Capture and manage subscriber account and service subscription information
- Estimate future usage based on past history
- Collect billable event data
- Generate and rate billing records
- Calculate, prepare and deliver bill
- Process and manage bill payment information and records
- Handle customer account inquires
- Manage debt and fraud

This Specification focuses only on various business scenarios on bandwidth-centric usage-based billing options.

### **B.3.4 Designing Usage-Based Billing Models**

In support of the offering of different class of services is a new set of billing processes, which are based on the accounting of actual usage of subscribed service by each subscriber calculated by the associated fee structures.

There are several alternatives to implementing usage-based billing. The following offers a few examples:

- Billing Based on an Average Bandwidth Usage.

**The average bandwidth usage is defined as the total bytes transmitted divided by the billing period.**

- Billing Based on Peak Bandwidth Usage.

**The peak bandwidth usage is the highest bandwidth usage sample during the entire billing period.**

**Each usage sample is defined as the average bandwidth usage over a data collection period (typically 10 minutes).**

Since it is usually the peak usage pattern that creates the highest possibility of access problems for the cable operator, therefore it is reasonable to charge for such usage. One scheme of peak usage billing is called “95 percentile billing”. The process is as follows -- at the end of each billing period, the billing software examines the usage records of each subscriber and it “throws away” the top five percent of usage records of that period, then charge the subscriber on the next highest bandwidth usage.

- “Flat Monthly Fee” Plus Usage Billing Based on the Class of Service Subscribed.

**Any usage beyond the minimum guaranteed bandwidth for that particular subscriber service class is subject to an extra charge based on the number of bytes transmitted.**

- Billing for “On Demand” Service

**This special billing process is to support the “On Demand” Service offering described above.**

## **Appendix C.    Propose Account Management MIB**

The account management MIB will be specified in this section by the ECR/ECO/ECN process.



## **Appendix D. SNMPv2c INFORM Request Definition for Subscriber Account Management (SAM)**

The INFORM Request definition of account management will be specified in this section by the ECR/ECO/ECN process.

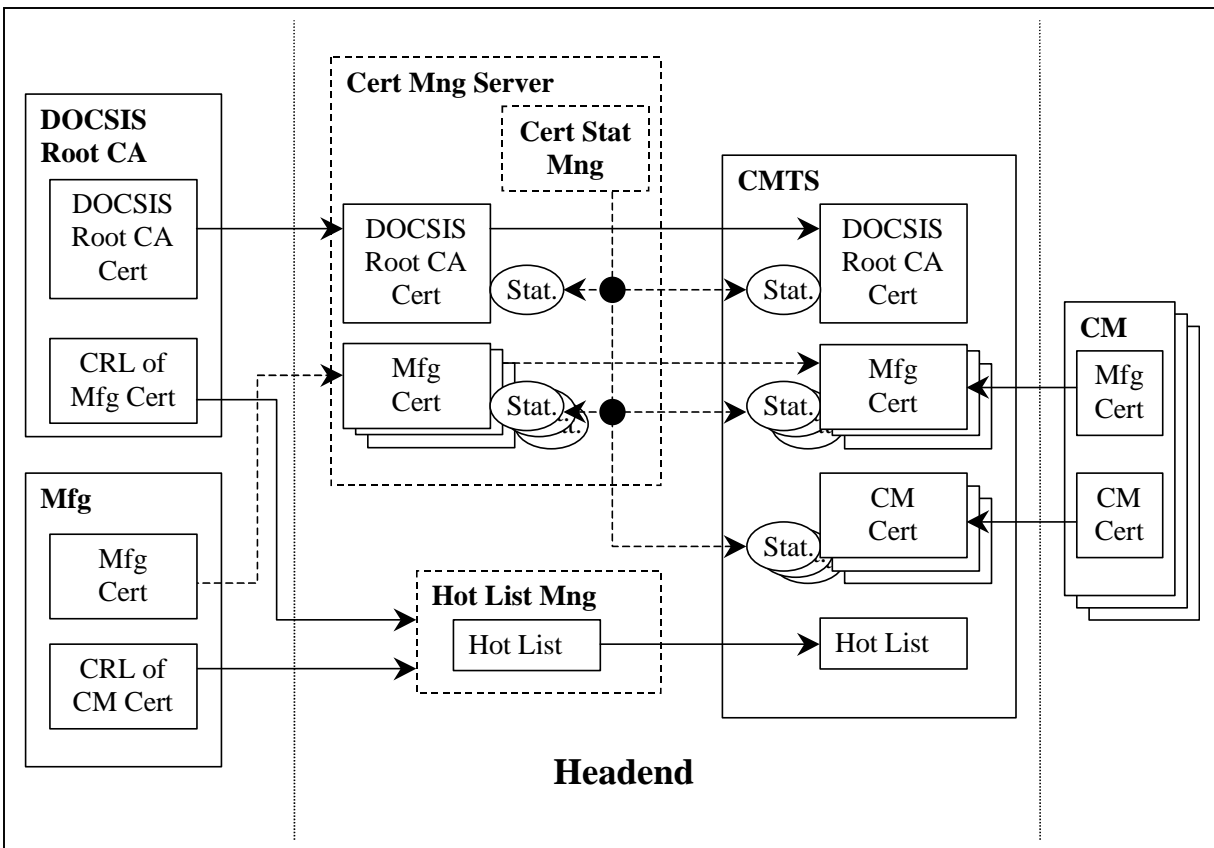
This page intentionally blank

## Appendix E. Summary of the CM Authentication and the Code File Authentication

The purpose of this appendix is to provide the overview of the two authentication mechanisms defined by BPI+ specification [SP-BPI+-I03] and also to provide a example of the responsibility assignment for actual operation but not to add any new requirements for the CMTS or the CM. Please refer BPI+ specification [SP-BPI+-I03] regarding the requirement for the CMTS and the CM.

### E.1 Authentication of the DOCSIS 1.1 compliant CM

If the CMTS is compliant to the DOCSIS 1.1/BPI+ and a DOCSIS 1.1 compliant CM is provisioned to run BPI+ by the CM configuration file, the CMTS authenticate the CM during the CM initialization by verifying the CM certificate and the manufacturer CA certificate. These certificates are contained in Auth Info message and Auth Request message separately and sent from the CM to the CMTS just after the CM registration. Only the CM with the valid certificates will be authorized by the CMTS and become ready to forward the user traffic. Note that this CM authentication won't be applied if the CMTS and/or the CM is not compliant to BPI+, or the CM is not provisioned to run BPI+.



### **E.1.1. Responsibility of the DOCSIS Root CA**

The DOCSIS Root CA is responsible for the following:

- Store the DOCSIS Root private key in secret.
- Maintain the DOCSIS Root CA certificate.
- Issue the manufacturer CA certificates signed by the DOCSIS Root CA.
- Maintain the CRL of the manufacturer CA.
- Provide the operators with the CRL.

It is not yet decided whether a manufacturer CA certificate signed by the DOCSIS Root CA is provided to the CM manufacturer before applying for the CableLabs' certification process or after achieving the certified status.

### **E.1.2 Responsibility of the CM manufacturers**

The CM manufacturers are responsible for the following:

- Store the manufacturer CA private key in secret,
- Maintain the manufacturer CA certificate. The manufacturer CA certificate is usually signed by the DOCSIS Root CA but can be self-signed until the DOCSIS Root CA issues it based on the CableLabs policy.
- Issue the CM certificates,
- Put the manufacturer CA certificate in the CM's software,
- Put each CM certificate in the CM's permanent, write-once memory.
- Provide the operators with the hot list of the CM certificate. The hot list may be in the CRL format. However, the detail of the format and the way of delivery are TBD.

### **E.1.3 Responsibility of the operators**

The operators are responsible for the following:

- Maintain that the CMTS(s) have an accurate date and time. If a CMTS has a wrong date or time, the invalid certificate may be authenticated or the valid certificate may not be authenticated.
- Put the DOCSIS Root CA certificate in the CMTS during the CMTS provisioning using BPI+ MIB or the CMTS's proprietary function. The operator may have a server to manage this certificate for one or more CMTS(s).
- Put the manufacturer CA certificate(s) in the CMTS during the CMTS provisioning using BPI+ MIB or the CMTS's proprietary function (optional). The operator may have a server to manage this certificate for one or more CMTS(s).

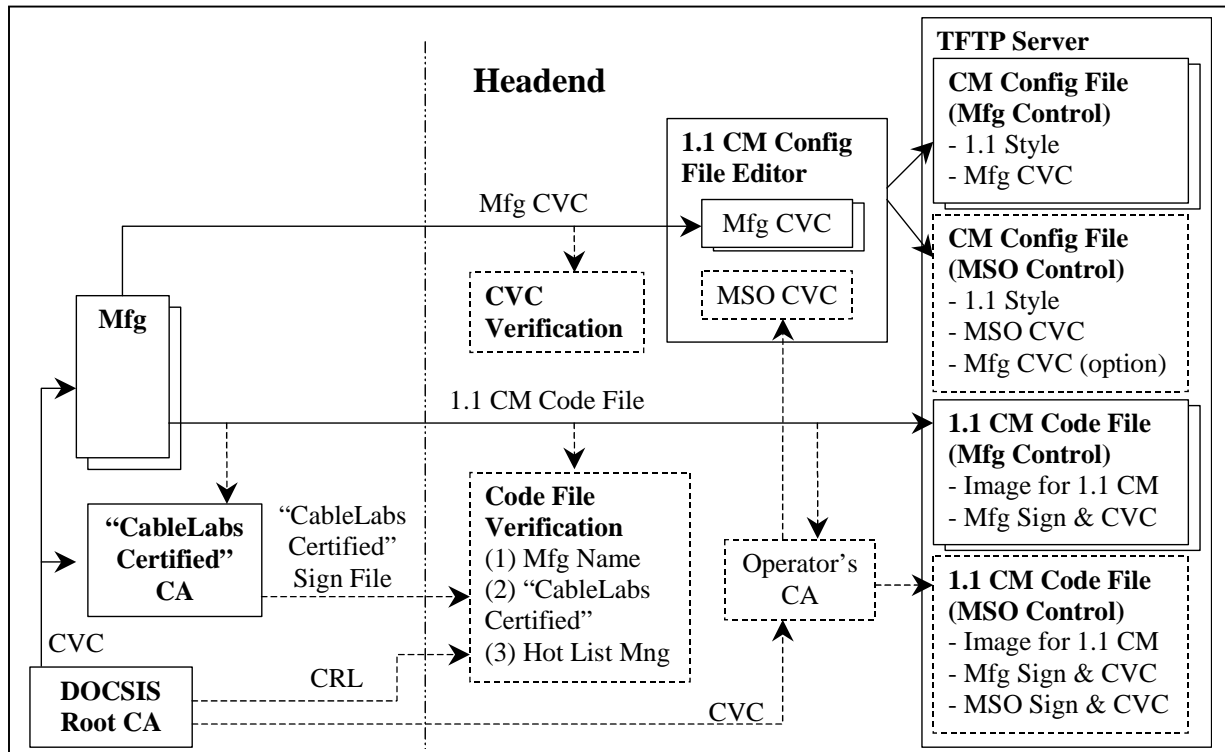
- Maintain the status of the certificates in the CMTS(s) if desired using BPI+ MIB or the CMTS's proprietary function (optional). The operator may have a server to manage all the status of the certificates recorded in one or more CMTS(s).

The operator may have a server to manage the DOCSIS Root CA certificate, manufacturer CA certificate(s) and also the status of the certificates recorded in one or more CMTS(s).

- Maintain the hot list for the CMTS based on the CRLs provided by the DOCSIS Root CA and the CM manufacturers (optional). The operator may have a server to manage the hot list based on the CRLs provided by the DOCSIS Root CA and manufacturer CAs. The CMTS may have a function to automatically download the DOCSIS Root CA certificate and the CRLs via the Internet or other method. The DOCSIS Root CA or CableLabs is likely to put the DOCSIS Root CA on their Web or TFTP server in order to let the operators (or the CMTS on behalf of the operator) download it but this is not yet decided.

## E.2 Authentication of the code file for the DOCSIS 1.1 compliant CM

When the DOCSIS 1.1/BPI+ compliant CM downloads the code file from TFTP server, the CM must always authenticate the code file as defined in the appendix D of [SP-BPI+-I03] regardless of whether the CM is provisioned to run BPI+, BPI or none of them by the CM configuration file. The CM installs the new image and restart using it only if the CVC(s) and the signature(s) in the code file are verified. If the authentication fails because of the invalid CVC(s) or signature(s) in the code file, the CM rejects the code file downloaded from the TFTP server and continues to operate using the current code. The CM accepts the order of the software downloading via the CM configuration file or the MIB only if the CM is properly initialised by the CVC(s) in the CM configuration file. In addition to the code file authentication by the CM, the operators may authenticate the code file before they put it on the TFTP sever. The following figure shows the summary of these mechanisms.



### **E.2.1 Responsibility of the DOCSIS Root CA**

The DOCSIS Root CA is responsible for the following:

- Store the DOCSIS Root private key in secret,
- Maintain the DOCSIS Root CA certificate, and
- Issue the code verification certificates (CVCs) for the CM manufacturers, for the operators, and for "CableLabs Certified(TM)".
- May maintain the CRL of the CVCs and provide it with the operators but not yet decided.

### **E.2.2 Responsibility of the CM manufacturer**

The CM manufacturers are responsible for the following:

- Store the manufacturer CVC private key in secret,
- Put the DOCSIS Root CA certificate in the CM's software,
- Maintain the manufacturer CVC. (Current BPI+ specification only allows the CVC signed by the DOCSIS Root CA and does not accept the self-signed CVC.)
- Generate the code file with the manufacturer's CVC and signature, and
- Provide the operators with the code file and the manufacturer CVC,

### **E.2.3 Responsibility of CableLabs**

CableLabs is responsible for the following:

- Store the "CableLabs Certified(TM)" CVC private key in secret,
- Maintain the "CableLabs Certified(TM)" CVC signed by the DOCSIS Root CA.
- Issue the "CableLabs Certified(TM)" signature file for the DOCSIS 1.1 CM code file certified by CableLabs.

### **E.2.4 Responsibility of the operators**

The operator has the following responsibility and options:

- Check the manufacturer of the code file by verifying the manufacturer's CVC and signature in the code file provided by the CM manufacturer before the operator load the code file on the TFTP server (optional). The code file may be rejected and won't be loaded on the TFTP server if the unexpected manufacturer signs it or the CVC and/or the signature in it are invalid.
- Check if the code file provided by the CM manufacturer is "CableLabs Certified(TM)" by verifying the "CableLabs Certified(TM)"'s CVC and signature in the "CableLabs Certified(TM)" signature file against the code file before the operator load the code file on the TFTP server (optional). CableLabs is likely to post all the "CableLabs Certified(TM)" signature files and also the corresponding certified code files on the web or FTP server while this is not yet decided. Whether this information is open to only the CableLabs members, all the operators, all the vendors, or public is not yet decided

- Operate the operator CA by storing the operator CA private key in secret and maintaining the operator's (co-signer) CVC issued by the DOCSIS Root CA (optional).
- Generate the MSO-controlled code file by adding the operator's CVC and signature to the original code file provided by the CM manufacturer (optional).
- Check if the CVC provided by the CM manufacturer is valid (optional).
- Put the appropriate CVC(s) in the CM configuration file. In case that the original code file is to be downloaded to the CMs, the CM configuration file must contain the valid CVC from the CM's manufacturer. In case that the operator-controlled code file is to be downloaded, the CM configuration file must contain the valid CVC of the operator and may contain the valid CVC from the CM manufacturer. If there is no CVC in the CM configuration file or all the CVC(s) in the CM configuration file is invalid, the CM won't accept any order of the software downloading via the CM configuration file and the MIB. Note that the DOCSIS 1.1 compliant CM may be registered and authorized by the CMTS and becomes operational regardless of whether the CM configuration file contains the valid CVC(s).

## Appendix F. Events for Notification

PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	EVENT DETAILS	ERROR CODE SET	Trap OID (docsdev.2.xxx) where xxx is (below) is used to identify a specific event
			DHCP and TOD FAILED before registration		D00	
Init	Critical		DHCP FAILED - Discover sent, no offer received		D01.0	1140850944
Init	Critical		DHCP FAILED - Request sent, No response		D02.0	1140851200
Init	Critical		DHCP FAILED - Requested Info not supported.		D03.0	1140851456
Init	Critical		DHCP FAILED - Response doesn't contain ALL the valid fields as describe in the RFI spec Appendix D		D03.1	1140851457
			DOWNSTREAM ACQUISITION FAILED		T00	
Init	Critical		SYNC Timing Synchronization failure, Failed to acquire QAM/QPSK symbol timing, Error stats Retry #'s		T01.0	1409286400
Init	Critical		SYNC Timing Synchronization failure, Failed to acquire FEC framing. Error stats Retry #'s # of bad frames		T02.0	1409286656
Init	Critical		SYNC Timing Synchronization failure, Acquired FEC framing. Failed to acquire MPEG2 Sync. Retry #'s		T02.1	1409286657
Init	Critical		SYNC Timing Synchronization failure, Failed to acquire MAC framing. Error stats Retry #'s Of bad frames		T03.0	1409286912
Init	Critical		SYNC Timing Synchronization failure, Failed to receive MAC SYNC frame within time-out period.		T04.0	1409287168
Init	Critical		SYNC Timing Synchronization failure, Loss of Sync. (Missed 5 in a row, after having SYNC'd at one time)		T05.0	1409287424
			FAILED TO OBTAIN UPSTREAM PARAMETERS		U00	
Init	Critical		No UCD's Received. Timeout.		U01.0	1426063616
Init	Critical		UCD invalid or channel unusable.		U02.0	1426063872

PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	EVENT DETAILS	ERROR CODE SET	Trap OID (docsdev.2.xxx) where xxx is (below) is used to identify a specific event
Init	Critical		UCD, & SYNC valid, NO MAPS for this channel		U04.0	
Init	Critical		US channel wide parameters not set before Burst Descriptors.		U06.0	1426064896
			RANGING FAILED : RNG-REQ RANGING REQUEST		R00	
Init	Critical		No Maintenance Broadcasts for Ranging opportunities received T2 time-out.		R01.0	1375731968
Init	Critical		Received Response to Broadcast Maintenance Request, But no Unicast Maintenance opportunities received. T4 timeout.		R04.0	1375732736
			RANGING FAILED : RNG-REQ RANGING RESPONSE			
Init	Critical		No Ranging Response received, T3 time-out.		R02.0	1375732224
Init	Critical		Ranging Request Retries exhausted		R03.0	1375732480
Init	Critical		Started Unicast Maintenance Ranging no Response received. T3 time-out.		R05.0	1375732992
Init	Critical		Unicast Maintenance Ranging attempted. No response. Retries exhausted.		R06.0	1375733248
Init	Critical		Unicast Ranging Received Abort Response. Re-initializing MAC.		R07.0	1375733504
			ToD FAILED			
Init	Error		Time of Day Request sent no Response received.		D04.1	1140851713
Init	Error		Time of Day Response received but invalid data/format.		D04.2	1140851714
			DHCP and TOD FAILED before registration		D00	
Init	Critical		TFTP failed, request sent, No Response/No server.		D05.0	1140851968
Init	Critical		TFTP failed, configuration file NOT FOUND.		D06.0	1140852224
Init	Critical		TFTP Failed, OUT OF ORDER packets.		D07.0	1140852480
Init	Critical		TFTP complete, but failed Message Integrity check (MIC)		D08.0	1140852736



PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	EVENT DETAILS	ERROR CODE SET	Trap OID (docsdev.2.xxx) where xxx is (below) is used to identify a specific event
			REGISTRATION FAILED (REG- REQ REGISTRATION REQUEST)		I10	
Init	Critical	Warning	Registration Failed, Service not available	Unrecognized configuration setting	I04.1	1224737793
	Critical	Warning		Temporarily unavailable	I04.2	1224737794
	Critical	Warning		Permanent.	I04.3	1224737795
Init	Critical	Warning	Registration Failed - Registration request	Invalid MAC header.	I101.0	1224762624
	Critical	Warning		Invalid SID, not in use.	I102.0	1224762880
	Critical	Warning		Required TLV's not present.	I104.0	1224763392
Init	Critical	Warning	Registration Failed, Bad Downstream Frequency	Format Invalid	I105.0	1224763648
	Critical	Warning		Not in use	I105.1	1224763649
	Critical	Warning		Not multiple of 62500Hz	I105.2	1224763650
Init	Critical	Warning	Registration Failed, Bad Upstream Channel	Invalid, unassigned	I106.0	1224763904
	Critical	Warning		Change followed with (RE-) Registration REQ.	I106.1	1224763905
Init	Critical	Warning	Registration Failed, Network Access configuration has invalid parameter.		I108.0	1224764416
Init	Critical	Warning	Registration Failed, Bad Class of Service	configuration is invalid.	I109.0	1224764672
	Critical	Warning		Service ID unsupported	I110.0	1224764928
	Critical	Warning		Service ID invalid or out of range.	I111.0	1224765184
Init	Critical	Warning	Registration Failed, Bad Max Downstream Bit	configuration is invalid format	I112.0	1224765440
	Critical	Warning		configuration setting is unsupported.	I112.1	1224765441
Init	Critical	Warning	Registration Failed,Bad Max Upstream Bit Rate	Configuration setting invalid format	I113.0	1224765696

PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	EVENT DETAILS	ERROR CODE SET	Trap OID (docsdev.2.xxx) where xxx is (below) is used to identify a specific event
	Critical	Warning		Configuration setting unsupported	I113.1	1224765697
Init	Critical	Warning	Registration Failed,Bad Upstream Priority configuration	invalid format.	I114.0	1224765952
	Critical	Warning		setting out of range.	I114.1	1224765953
Init	Critical	Warning	Registration Failed,Bad Guaranteed Min Upstream Channel Bit rate configuration setting	invalid format.	I115.0	1224766208
	Critical	Warning		exceeds Max Upstream Bit rate.	I115.1	1224766209
	Critical	Warning		out of range.	I115.2	1224766210
Init	Critical	Warning	Registration Failed,Bad Max Upstream Channel Transmit Burst configuration setting	invalid format.	I116.0	1224766464
	Critical	Warning		out of range	I116.1	1224766465
Init	Critical	Warning	Registration Failed,Modem Capabilities configuration setting invalid format.		I117.0	1224766720
Init	Critical	Warning	Registration Failed, Config file parameters outside the range e.g. # of CPE's given in config file more than the Spec.		I118.0	1224766976
			VERSION 1.1 SPECIFIC REG-REQ REGISTRATION REQUEST		I200.0	
Init	Critical	Warning	DOCSIS 1.1 Registration rejected	unspecified reason.	I201.0	1224788224
	Critical	Warning		unrecognized configuration setting.	I201.0	1224788224
	Critical	Warning		temporary no resource.	I201.2	1224788226
	Critical	Warning		permanent administrative.	I201.3	1224788227
	Critical	Warning		required parameter not present.	I201.4	1224788228
	Critical	Warning		header suppression setting not supported.	I201.5	1224788229
			REG-RSP REGISTRATION RESPONSE		I00.0	
Init	Critical	Warning	Registration RESP Bad.	invalid format or not recognized.	I01.0	1224737024
	Critical	Warning		bad SID.	I03.0	1224737536

PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	EVENT DETAILS	ERROR CODE SET	Trap OID (docsdev.2.xxx) where xxx is (below) is used to identify a specific event
Init	Critical	Warning	Registration RESP not received.		I02.0	1224737280
			REG-ACK REGISTRATION ACKNOWLEDGEMENT		I300.0	
Init	Critical	Warning	Registration aborted no REG-ACK.		I301.0	1224813824
			TLV-11 ERRORS		I400.0	
Init	Error		TLV-11 error	CM SNMP Object(s) ignored	I401.0	1224839424
SW Upgrade	Information		SW download initiated	NMS_Initiated <IP address>, <TFTP server address>, < Filename>	E101.0	1157653760
	Information			CFG_File_Initiated <TFTP server address>, < Filename>	E102.0	1157654016
SW Upgrade	Error		SW download Failed	Max retry exceeded	E103.0	1157654272
SW Upgrade	Error		SW upgrade Error - before sucessful TFTP	Server not present <TFTP server addr>	E104.0	1157654528
	Error			File not present <Filename>	E105.0	1157654784
	Error			Tftp Max retry exceeded	E106.0	1157655040
SW Upgrade	Error		SW upgrade Error - after sucessful TFTP	Incompatible file	E107.0	1157655296
				File corruption	E108.0	1157655552
SW Upgrade	Error		Disruption during SW download.	Power lost	E109.0	1157655808
				RF removed.	E110.0	1157656064
SW Upgrade	Notice		SW download Successful	NMS_Initiated <IP address>, <TFTP server address>, < Filename>	E111.0	1157656320
				CFG_File_Initiated <TFTP server address>, < Filename>	E112.0	1157656576
DHCP			DHCP renewal failure		D100.0	
	Error			Renew sent, No response	D101.0	1140876544
	Error			Rebind Sent, No response	D102.0	1140876800
DHCP	Error		Invalid DHCP Options	Renew	D103.0	1140877056

PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	EVENT DETAILS	ERROR CODE SET	Trap OID (docsdev.2.xxx) where xxx is (below) is used to identify a specific event
	Error			Rebind	D104.0	1140877312
				DYNAMIC SERVICES	S00	
DYNAMIC SERVICES	Error	Warning	Dynamic Service requests - Service add rejected	unspecified reason	S01.0	1392509184
	Error	Warning		unrecognized configuration setting	S01.1	1392509185
	Error	Warning		permanent administrative	S01.3	1392509187
	Error	Warning		required parameter not present	S01.4	1392509188
	Error	Warning		HMAC authentication failure	S01.7	1392509191
DYNAMIC SERVICES	Error	Warning	Dynamic Service requests - Service change rejected	unspecified reason	S02.0	1392509440
	Error	Warning		unrecognized configuration setting	S02.1	1392509441
	Error	Warning		requestor not owner of service flow	S02.4	1392509444
	Error	Warning		service flow not found	S02.5	1392509445
	Error	Warning		HMAC authentication failure	S02.8	1392509448
DYNAMIC SERVICES	Error	Warning	Dynamic Service requests - Service delete rejected	unspecified reason	S03.0	1392509696
	Error	Warning		service flow not found.	S03.2	1392509698
	Error	Warning		HMAC authentication failure	S03.3	1392509699
			DYNAMIC SERVICE RESPONSES			
DYNAMIC SERVICES	Error	Warning	Service add response rejected invalid transaction ID		S101.0	1392534784
DYNAMIC SERVICES	Error	Warning	Service change response rejected invalid transaction ID.		S102.0	1392535040
DYNAMIC SERVICES	Error	Warning	Service delete response rejected invalid transaction ID		S103.0	1392535296
			DYNAMIC SERVICE ACKNOWLEDGEMENTS			

PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	EVENT DETAILS	ERROR CODE SET	Trap OID (docsdev.2.xxx) where xxx is (below) is used to identify a specific event
DYNAMIC SERVICES	Error	Warning	Dynamic Service Acknowledgements - Service add	ACK rejected invalid transaction ID	S201.0	1392560384
	Error	Warning		aborted no ACK	S201.1	1392560385
DYNAMIC SERVICES	Error	Warning	Dynamic Service Acknowledgements -Service change	ACK rejected invalid transaction ID	S202.0	1392560640
			CM CONFIGURATION FILE		B100	
Init (BPI+)	Warning	Error	Missing BP Configuration Setting TLV		B101.0	1107322112
Init (BPI+)	Warning	Error	Invalid BP Configuration Setting Value		B102.0	1107322368
			CERTIFICATE VERIFICATION		B200	
Init (BPI+)	Error	Error	CM Certificate Format Error		B201.0	1107347712
Init (BPI+)	Error	Error	Manufacture CA Certificate Format Error		B202.0	1107347968
Init (BPI+)	Error	Error	CM Certificate Self-Verification Failure		B203.0	1107348224
			AUTH FSM		B300	
BPKM	Warning	Error	Auth Reject -- No Information		B301.2	1107373314
BPKM	Warning	Error	Auth Reject -- Unauthorized CM		B301.3	1107373315
BPKM	Warning	Error	Auth Reject -- Unauthorized SAID		B301.4	1107373316
BPKM	Warning	Error	Auth Reject -- Permanent Authorization Failure		B301.8	1107373320
BPKM	Warning	Error	Auth Invalid -- No Information		B302.2	1107373570
BPKM	Warning	Error	Auth Invalid -- Unauthorized CM		B302.3	1107373571
BPKM	Warning	Error	Auth Invalid -- Unsolicited		B302.5	1107373573
BPKM	Warning	Error	Auth Invalid -- Invalid Key Sequence Number		B302.6	1107373574
BPKM	Warning	Error	Auth Invalid -- Message (Key Request) Authentication Failure		B302.7	1107373575
BPKM	Warning	Error	Unsupported Crypto Suite		B303.0	1107373824
			EVENT BETWEEN AUTH & TEK FSM		B400	
BPKM	Informational		Authorized		B401.0	1107398912

PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	EVENT DETAILS	ERROR CODE SET	Trap OID (docsdev.2.xxx) where xxx is (below) is used to identify a specific event
BPKM	Informational		Auto Pend		B402.0	1107399168
BPKM	Informational		Auth Comp		B403.0	1107399424
BPKM	Informational		Stop		B404.0	1107399680
			TEK FSM		B500	
BPKM	Warning	Error	Key Reject -- No Information		B501.2	1107424514
BPKM	Warning	Error	Key Reject -- Unauthorized SAID		B501.3	1107424515
BPKM	Warning	Error	TEK Invalid -- No Information		B502.3	1107424771
BPKM	Warning	Error	TEK Invalid -- Invalid Key Sequence Number		B502.6	1107424774
			SA MAP FSM		B600	
Dynamic SA	Informational		SA Map State Machine Started		B601.0	1107450112
Dynamic SA	Warning	Error	Unsupported Crypto Suite		B602.0	1107450368
Dynamic SA	Error		Map Request Retry Timeout		B603.0	1107450624
Dynamic SA	Notice		Unmap		B604.0	1107450880
Dynamic SA	Warning	Error	Map Reject -- Not Authorized for Requested Dwonstream Traffic Flow (EC=7)		B605.9	1107451145
Dynamic SA	Warning	Error	Map Reject -- Dwonstream Traffic Flow Not Mapped to BPI+ SAID (EC=8)		B605.10	1107451137
Dynamic SA	Warning	Error	Mapped to Existing SAID		B606.0	1107451392
Dynamic SA	Warning	Error	Mapped to New SAID		B607.0	1107451648
			VERIFICAITON OF CODE FILE		E200	
SW Upgrade	Error		Improper Code File Controls		E201.0	1157679360
SW Upgrade	Error		Code File Manufacturer CVC Validation Failure		E202.0	1157679616
SW Upgrade	Error		Code File Manufacturer CVS Validation Failure		E203.0	1157679872
SW Upgrade	Error		Code File Co-Signer CVC Validation Failure		E204.0	1157680128
SW Upgrade	Error		Code File Co-Signer CVS Validation Failure		E205.0	1157680384
SW Upgrade	Error		Improper Configuration File CVC Format		E206.0	1157680640
SW Upgrade	Error		Configuration File CVC Validation Failure		E207.0	1157680896

PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	EVENT DETAILS	ERROR CODE SET	Trap OID (docsdev.2.xxx) where xxx is (below) is used to identify a specific event
SW Upgrade	Error		Improper SNMP CVC Format		E208.0	1157681152

## **Appendix G. Trap Definitions for Cable Device**

The trap definition for cable device will be specified in this section by the ECR/ECO/ECN process.



## Appendix H. References

- [IETF2] draft-ietf-idmr-igmp-mib-13.txt, Keith M., Dino Farinacci, "Internet Group Management Protocol MIB", Jan 31, 2000
- [IETF3] draft-ietf-ipcdn-igmp-mib-00.txt, H. Abramson, "Docsis 1.1 IGMP MIB", June 1999
- [IETF4] draft-ietf-ipcdn-qos-mib-01.txt, Mike Patrick, John Harvey, "Data Over Cable System Quality of Service Management Information Base", June 25, 1999
- [IETF5] Proposed Standard RFC version of IGMP MIB, "**draft-ietf-idmr-igmp-mib-02.txt**"
- [IETF6] Proposed Standard RFC version of BPI+ MIB, "**draft-ietf-ipcdn-bpiplus-02.txt**"
- [IETF7] Proposed Standard RFC version of USB MIB, "**draft-ietf-xxxx-xxxx-xxxx-00.txt**"
- [IETF8] Proposed Standard RFC version of BPI MIB, "**draft-ietf-ipcdn-bpi-01.txt**"
- [IETF9] Proposed Standard RFC version of Customer Management MIB, "**draft-ietf-subscriber-mib-01.txt**"
- [MCNS1] MCNS Cable Modem Termination System - Network-Side Interface Specification SP-CMTS-NSI-I01-960702
- [MCNS2] MCNS Cable Modem to Customer Premise Equipment Interface Specification SP-CMCI-D02c-991015
- [MCNS 3] MCNS Operations Support System Framework TR-OSSF-W08-961016
- [MCNS 4] MCNS Data Over Cable Services Cable Modem Telephony Return Interface Specification SP-CMTRI-I01-970804
- [MCNS 5] MCNS Data Over Cable Services Cable Modem Radio Frequency Interface Specification SP-RFIV1.1-I04-000331
- [MCNS 6] MCNS Data Over Cable Services Security Specification SP-SS-I01-970506
- [RFC-1157] Schoffstall, M., Fedor, M., Davin, J. and Case, J., A Simple Network Management Protocol (SNMP), IETF RFC-1157, May, 1990
- [RFC-1213] K. McCloghrie and M. Rose. Management Information Base for Network Management of TCP/IP-base internets: MIB-II, IETF RFC-1213, March, 1991

- [RFC-1224] L. Steinberg., Techniques for Managing Asynchronously Generated Alerts, IETF RFC-1224, May, 1991
- [RFC-1493] E. Decker, P. Langille, A. Rijssinghani, and K. McCloghrie., Definitions of Managed Objects for Bridges, IETF RFC-1493, July, 1993
- [RFC-1901] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.
- [RFC-1903] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1903, January 1996.
- [RFC-1905] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.
- [RFC-1906] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, January 1996
- [RFC-1907] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1907, January 1996.
- [RFC-2011] K. McCloghrie, "Category: Standards Track SNMPv2 Management Information Base for the Internet Protocol using SMIV2", November 1996
- [RFC-2013] K. McCloghrie, "Category: Standards Track SNMPv2 Management Information Base for the User Datagram Protocol using SMIV2", November 1996
- [RFC-2132] S. Alexander, R. Droms. DHCP Options and BOOTP Vendor Extensions. IETF RFC-2132. March, 1997.
- [RFC-2233] K. McCloghrie, F. Kastenholz, "The Interfaces Group MIB using SMIV2 ", November 1997
- [RFC-2358] J. Flick, J. Johnson, "Definitions of Managed Objects for the Ethernet-like Interface Types", June 1998
- [RFC-2570] J. Case, R. Mundy, D. Partain, B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework", April 1999
- [RFC-2571] Harrington, D., Presuhn, R. and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2571, April 1999.
- [RFC-2572] Case, J., Harrington, D., Presuhn, R. and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2572, April 1999
- [RFC-2573] Levi, D., Meyer, P. and B. Stewart, "SNMP Applications", RFC 2573, April 1999.

- [RFC-2574] Blumenthal, U. and B. Wijnen, "The User-Based Security Model for Version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2574, April 1999
- [RFC-2575] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model for the Simple Network Management Protocol (SNMP)", RFC 2575, April 1999
- [RFC-2576] R. Frye, D. Levi, S. Routhier, B. Wijnen, "Coexistence between Version 1, Version 2, and Version 3 of the Internet-Standard and Network Management Framework", RFC 2576, March 2000.
  
- [RFC-2578] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999
- [RFC-2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999
- [RFC-2580] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999
  
- [RFC-2669] M. St. Johns, "DOCSIS Cable Device MIB Cable Device Management Information Base for DOCSIS compliant Cable Modems and Cable Modem Termination Systems", August 1999
- [RFC-2670] M. St. Johns, "Radio Frequency (RF) Interface Management Information Base for MCNS/DOCSIS compliant RF interfaces", August 1999
  
- [RFC-2786] stjohns-snmpv3-dhkeychange-mib-01.txt, Michael C. StJohns, "Diffie-Helman USM Key MIB August 1999 Diffie-Helman USM Key Management Information Base and Textual Convention", Aug 6, 1999

This page intentionally blank

## Appendix I. Acknowledgements

The following contributors deserve genuine gratitude for their efforts in the development of the OSSI 1.1 specification.

Pak Siripunkaw of MediaOne

Taft Singletary of Cox Communications

Jason Schnitzer of Shaw

Mike St. Johns of @Home

Asha Hegde of Cisco

Daniel Chuang of 3Com

Minnie Lu of Cisco

Bill Yost of TCE

Kaz Ozawa of Toshiba

Bob Himlin of TurboNet

Douglas Jones of MediaOne

Tasheer Syed of TCE

Benjamin Dolnik of 3Com

Randy Demuynck of Ericsson

Raymond Hou of Amplifynet

Fred Kiremidjian of Amplifynet

Adam Parmelee of Terayon

Dan Smith of Broadband Access Systems, Inc.

Pavaz Kokan of TCE

CableLabs and the cable industry as a whole are grateful to these individuals and organizations for their contributions. Their diligent work and professional approach should be commended and their continued enthusiasm will be invaluable as the OSSI specification evolves.